AFIT/DS/ENS/98-02

FEATURE SALIENCY IN
ARTIFICIAL NEURAL NETWORKS WITH
APPLICATION TO MODELING WORKLOAD

DISSERTATION

Kelly A. Greene
Captain, USAF

AFIT/DS/ENS/98-02

19990108 042

AFIT/DS/ENS/98-02

FEATURE SALIENCY IN ARTIFICIAL NEURAL NETWORKS WITH APPLICATION TO MODELING WORKLOAD

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Kelly A. Greene
Captain, USAF

December 1998

AFIT/DS/ENS/98-02

FEATURE SALIENCY IN ARTIFICIAL NEURAL NETWORKS
WITH APPLICATION TO MODELING WORKLOAD

Kelly A. Greene
Captain, USAF

Approved: Date:

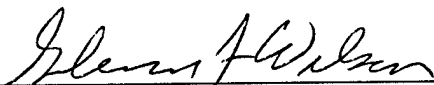_____   _____
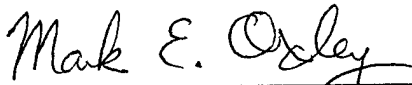Dr. Kenneth W. Bauer, Jr. (Chairman)   8 DEC 98

_____   _____
Dr. Matthew Kabrisky   6 DEC 98

_____   _____
Dr. Steven K. Rogers   8 DEC 98

_____   _____
Dr. Glenn F. Wilson   8 Dec 98

_____   _____
Dr. Mark E. Oxley (Dean's Representative)   8 Dec 98

Accepted:

_____
Dr. Robert A. Calico
Dean, School of Engineering

ii

Acknowledgments

This dissertation represents three years of very hard work. In many ways, these past three years were the toughest yet most rewarding years of my life. This dissertation would never have reached completion without the constant support of my research committee: Dr. Kenneth W. Bauer, Jr., Dr. Matthew Kabrisky, Dr. Steven K. Rogers, and Dr. Glenn F. Wilson. First and foremost, I would like to thank Dr. Bauer for chairing my research committee and bringing me back down to Earth every now and then. This dissertation would not have been possible without Dr. Bauer because the signal-to-noise ratio (SNR) screening method and its SNR saliency measure, both of which are used extensively in this dissertation, were his ideas. Dr. Kabrisky gets all of the credit for my first exposure to artificial neural networks (ANN) in 1991. It was his enthusiasm for ANNs that inspired me to pursue a Ph.D. to gain further understanding of ANNs. I praise Dr. Rogers for always providing theoretical insights that, at first, were way over my head and always pushing me to think things through. I am grateful to Dr. Wilson for always suggesting an interesting application. I would also like to thank Dr. Mark E. Oxley for being an outstanding Dean's Representative and for helping me "clean up" and "clear up" my notation.

I am thankful for the funding provided by Dr. John F. Tangney of the Air Force Office of Scientific Research (AFOSR). I also want to thank the engineers and scientists at the Air Force's Flight Psychophysiological Laboratory for providing all of the workload data used in this dissertation. I thank Joel Coons for providing outstanding computer support that, at times, went way beyond the call of duty.

I thank the gang over at the centrifuge and my fellow pilots at the Aero Club who

provided me with some adventurous and exhilarating experiences. Godspeed and may you always have a tailwind. I thank my volleyball teammates who provided an avenue to let off some steam. I am grateful to have had so many good friends here in Dayton to support me: Shelly Benson, Cynthia Fisher, Lisa Fitzgerald, and Claudia Kropas-Hughes. Our friendships are forever.

I am very grateful to H. Edward Shaffer and Jane G. Waymire, my parents. I am also grateful to William E. Waymire and Suzanne Shaffer. Thanks for listening and for your encouraging words. Of everyone, though, I thank Christopher D. Greene, my husband, for always supporting my *dream* and hence, my decision to accept the offer at AFIT to get a Ph.D.

Finally, I thank the United States Air Force for yet another opportunity. It's amazing the accomplishments one can achieve if one just tries - the only true failure in life is not trying. Aim high. The sky is *not* the limit.

Kelly A. Greene

This dissertation is dedicated to my "heroes" for it is my heroes who have provided a

burning desire in me to succeed at all that I do and a passion for the adventure that life

presents to us everyday.

Colonel Edwin E. "Buzz" Aldrin, Jr., Ph.D., United States Air Force
Apollo 11 Astronaut
Extraordinary Adventurer
1930-

Jerrie Cobb
America's True First Female Astronaut Candidate
One of the Mercury 13
Nobel Peace Prize Nominee
1931-

Amelia Earhart
Aviation Pioneer who opened the Gates of Aviation to Women
1897-1937

Table of Contents

x

xi

List of Figures

List of Tables

abs	absolute
AFB	Air Force Base
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
AGARD	Advisory Group for Aerospace Research and Development
AL	Armstrong Laboratory
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
Apr	April
ASME	American Society of Mechanical Engineers
ASRS	Aviation Safety Reporting System
ATC	Air Traffic Controller
Aug	August
avg	average
CA	California
CI	Confidence Interval
CLT	Central Limit Theorem
DC	District of Columbia
dB	decibel
Dec	December
DME	Distance Measuring Equipment
ECG	Electrocardiography
ed	editor
EEG	Electroencephalography
EKG	Electrocardiography
EMG	Electromyography
ENG	Graduate School of Engineering, Department of Electrical and Computer Engineering
ENS	Graduate School of Engineering, Department of Operational Sciences
EOG	Electro-oculography
EP	Evoked Potential
Err	error
FAA	Federal Aviation Administration
Feb	February
FFT	Fast-Fourier Transform
G-LOC	G-induced Loss of Consciousness
HECP	Human Effectiveness Directorate, Human Interface Technology Branch
Hz	Hertz
IID	Identically and Independently Distributed
ILS	Instrument Landing System
Jan	January
Jr.	Junior

Jun	June
Jul	July
K-L	Karhunen-Loève transformation
LA	Los Alamos
MA	Massachusetts
Mar	March
MD	Maryland
Min	minimum
MIT	Massachusetts Institute of Technology
MOE	Measure Of Effectiveness
MLP	Multi-Layer Perceptron
M.S.	Master of Science
MSE	mean squared error
NASA	National Aeronautics and Space Administration
NM	New Mexico
No	Number
Nov	November
NY	New York
OAI	Ohio Aerospace Institute
OH	Ohio
p.	page
PC	Principal Component, Personal Computer (Note: PC is used in two very distinct ways in this dissertation. However, no suitable substitute is appropriate in any either case.)
PCA	Principal Component Analysis
Ph.D.	Doctorate of Philosophy
pp.	pages
REG	Rheoencephalography
RNN	Recurrent Neural Network
Sep	September
SNR	Signal-to-Noise Ratio
TLX	Task Load Index
TR	Technical Report
TRACON	Terminal Radar Approach Control
trn	training
TX	Texas
UK	United Kingdom
USA	United States of America
USAF	United States Air Force
Vol	Volume
WAM	Workload Assessment Monitor
WSO	Weapon Systems Officer
XOR	Exclusive OR

α	Alpha EEG frequency band (see Table 3 or Table 4) or level of significance (Note: α is used in two very distinct ways in this dissertation. However, no suitable substitute is appropriate in any either case.)
α_1	Alpha EEG frequency band (see Table 4)
α_2	Alpha EEG frequency band (see Table 4)
β	Beta EEG frequency band (see Table 3) or user-defined parameter in Setiono-Liu penalty function (Note: β is used in two very distinct ways in this dissertation. However, no suitable substitute is appropriate in any either case.)
β_1	Beta EEG frequency band (see Table 4)
β_2	Beta EEG frequency band (see Table 4)
Φ	constant vector used in unifying relationship between Λ_i and τ_i^{v2}
χ^2	chi-square
χ_s^2	χ^2 test statistic
Δ	Delta EEG frequency band (see Table 3 and Table 4)
ΔCA_{test}	allowable maximum decrease in CA_{test}
$\in$	radius of a ball
$\in_1$	user-defined parameter in Setiono-Liu penalty function
$\in_2$	user-defined parameter in Setiono-Liu penalty function
Γ	partial derivative-based saliency measure over time
$\lambda_{i,i}$	diagonal element of Λ, eigenvalue associated with i^{th} eigenvector
Λ	eigenvalue matrix
Λ_i	partial derivative-based saliency measure for feature $i = 1,...,I$
Λ_{rank}	partial derivative-based saliency measure for feature with ranking *rank*
$\Lambda_{i,g}$	partial derivative-based saliency measure for feature $i = 1,...,I$ for training session $g = 1,...,G$
$\Lambda_{N,g}$	partial derivative-based saliency measure for injected noise feature N for training session $g = 1,...,G$
$\hat{\Lambda}_i$	partial derivative-based saliency measure with pseudo-sampling for feature $i = 1,...,I$
$\hat{\Lambda}_{rank}$	partial derivative-based saliency measure with pseudo-sampling for feature with ranking *rank*
$\overline{\hat{\Lambda}}_i$	observed average of $\hat{\Lambda}_{i,g}$

$\widehat{\Lambda}_{i,g}$	partial derivative-based saliency measure with pseudo-sampling for feature $i = 1,\dots,I$ for training session $g = 1,\dots,G$
$\overline{\widehat{\Lambda}}_{N}$	observed average of $\widehat{\Lambda}_{N,g}$
$\widehat{\Lambda}_{N,g}$	partial derivative-based saliency measure with pseudo-sampling for injected noise feature N for training session $g = 1,\dots,G$
η	learning rate
η_0	initial learning rate
$\eta(E)$	learning rate as a function of number of epochs
$\rho_{i,d}$	sample correlation between feature $i = 1,\dots,I$ and desired output d
θ	Theta EEG frequency band (see Table 3 and Table 4)
τ_i	weight-based saliency measure for feature $i = 1,\dots,I$
τ_{rank}	weight-based saliency measure for feature with ranking $rank$
$\overline{\tau}_I$	observed average of $\tau_{i,g}$
$\tau_{i,g}$	weight-based saliency measure for feature $i = 1,\dots,I$ for training session $g = 1,\dots,G$
$\overline{\tau}_N$	observed average of $\tau_{N,g}$
$\tau_{N,g}$	weight-based saliency measure for injected noise feature N for training session $g = 1,\dots,G$
τ_i^{v1}	first variant of τ_i
τ_{rank}^{v1}	first variant of τ_{rank}
$\tau_{i,g}^{v1}$	first variant of $\tau_{i,g}$
$\tau_{N,g}^{v1}$	first variant of $\tau_{N,g}$
τ_i^{v2}	second variant of τ_i
τ_{rank}^{v2}	second variant of τ_{rank}
τ_i^{v3}	third variant of τ_i
τ_{rank}^{v3}	third variant of τ_{rank}
T	total number of time steps
μ_{D_i}	expected difference between the saliency value for feature $i = 1,2,\dots,I$ and that for the injected noise feature N
μ_{Λ_i}	expected partial derivative-based saliency measure for feature $i = 1,\dots,I$
μ_{Λ_N}	expected partial derivative-based saliency measure for injected noise feature N
$\mu_{\widehat{\Lambda}_N}$	expected partial derivative-based saliency measure with pseudo-sampling for injected noise feature N

μ_{τ_N}	expected weight-based saliency measure for injected noise feature N
$\mu_{\tau_i^{v1}}$	first variant of expected weight-based saliency measure for feature $i = 1,\ldots,I$
$\mu_{\tau_N^{v1}}$	first variant of μ_{τ_N}
μ_{CA}	expected classification accuracy
$\mu_{CA_{test}}$	expected classification accuracy of the test set
$\mu_{CA_{train}}$	expected classification accuracy of the training set
$\mu_{CA_{valid}}$	expected classification accuracy of the validation set
$\mu\beta$	UltraBeta EEG frequency band (see Table 3)
$\mu\beta_1$	UltraBeta EEG frequency band (see Table 4)
$\mu\beta_2$	UltraBeta EEG frequency band (see Table 4)
a	activation in transfer function
a_i	rank assigned to feature $i = 1,\ldots,I$
A	attractor
b_i	rank assigned to feature $i = 1,\ldots,I$
$\mathbf{c}$	correction increment vector in Rosenblatt's perceptron
$\mathbf{C}$	covariance matrix
C3	EEG scalp location (see Figure 16)
C4	EEG scalp location (see Figure 16)
$C(i,d)$	sample covariance between input $i = 1,\ldots,I$ and desired output d
$\overline{CA}$	observed average classification accuracy
CA_{test}	test set classification accuracy
CA_{test}^i	test set classification accuracy without feature $i = 1,\ldots,I$
$\overline{CA}_{test}^i$	observed average of CA_{test}^i
CA_{test}^g	test set classification accuracy for training session $g = 1,\ldots,G$
$\overline{CA}_{test}$	observed average of CA_{test}^g
CA_{train}	training set classification accuracy
CA_{train}^g	training set classification accuracy for training session $g = 1,\ldots,G$
$\overline{CA}_{train}$	average observed classification accuracy of the training set
$\overline{CA}_{train}(17\,\text{features})$	$\overline{CA}_{train}$ with top 17 ranked features
$\overline{CA}_{train}(33\,\text{features})$	$\overline{CA}_{train}$ with all 33 features
$\overline{CA}_{train}(\text{Best}\,\overline{CA}_{valid})$	$\overline{CA}_{train}$ with combination of top ranked features that produces the best $\overline{CA}_{valid}$
CA_{valid}	validation set classification accuracy

CA^g_{valid}	validation set classification accuracy for training session $g = 1,\ldots,G$
$\overline{CA}_{valid}$	average observed classification accuracy of the validation set
$class_m$	ANN classification for exemplar $m = 1,\ldots,M$
Cz	EEG scalp location (see Figure 15)
d	desired output
$\overline{d}$	mean desired output
$d_f(A)$	fractal dimension of A
d_m	desired output for exemplar $m = 1,\ldots,M$
$\mathbf{d}_m$	desired output vector for exemplar $m = 1,\ldots,M$
$d_{k,m}$	desired value of output node $k = 1,\ldots,K$ for exemplar $m = 1,\ldots,M$
$\overline{D}_i$	observed average of $D_{i,g}$
$D_{i,g}$	difference between the saliency value for feature $i = 1,2,\ldots,I$ and that for the injected noise feature N for training session $g = 1,2,\ldots,G$
e	$e \approx 2.71828$, index for number of epochs (Note: e is used in two very distinct ways in this dissertation. However, no suitable substitute is appropriate in any either case.)
E	total number of epochs
$f(a)$	transfer function
$f_j(a)$	transfer function for hidden node $j = 1,2,\ldots,J$
$f_k(a)$	transfer function for output node $k = 1,2,\ldots,K$
$\dot{f}(a)$	first derivative of transfer function $f(a)$
$\dot{f}_j(a)$	first derivative of transfer function $f_j(a)$ for hidden node $j = 1,2,\ldots,J$
$\dot{f}_k(a)$	first derivative of transfer function $f_k(a)$ for output node $k = 1,2,\ldots,K$
$f_{k,\ell}$	number of exemplars that the ANN classified as belonging to class $k = 1,\ldots,4$ in confusion matrix $\ell = 1,2$
FP1	EEG scalp location (see Figure 16)
Fz	EEG scalp location (see Figure 15 or Figure 16)
g	index for training sessions
G	total number of training sessions
H_0	null hypothesis
H_a	alternate hypothesis
i	index for features, index for input nodes
I	total number of features, total number of input nodes
$I(EEG)$	effective value of an EEG signal
$I(Pulse)$	effective value of a rectangle pulse

$I(S)$	effective value of a signal S
j	index for hidden nodes
J	total number of hidden nodes
k	index for classes, index for output nodes
K	total number of classes, total number of output nodes
lag	index for time lags
ℓ	index for confusion matrices
L	total number of confusion matrices; total number of time lags
lim	limit
$\lin(a)$	linear transfer function
ln	natural logarithm so that $\ln(x) = \log_e(x)$
log	logarithm so that $\log(x) = \log_{10}(x)$
m	index for exemplars
M	total number of exemplars
M_{test}	total number of exemplars in the test set
M_{train}	total number of exemplars in the training set
M_{train}^{low}	total number of low workload exemplars in the training set
M_{valid}	total number of exemplars in the validation set
$\max(amp)$	maximum amplitude
$\max\left(amp_{\sin(w)}\right)$	maximum amplitude of sin wave $w = 1,2,\dots,5$
$\max[x_i(train,test)]$	maximum value of feature $i = 1,\dots,I$ in the training set and the test set
m_c	momentum constant
$\min[x_i(train,test)]$	minimum value of feature $i = 1,\dots,I$ in the training set and the test set
MSE_{test}	test set mean squared error
MSE_{train}	training set mean squared error
$MSE_{train,m}$	training set mean squared error for example $m = 1,\dots, M_{train}$
MSE_{valid}	validation set mean squared error
N	injected noise feature
N1	component of EP (see Figure 17)
N2	component of EP (see Figure 17)
$N(A,\in)$	smallest number of balls with radius $\in$ required to cover A
O1	EEG scalp location (see Figure 15)
p	dimensional space used in Grassberger and Procaccia method for calculating $d_f(A)$
$\mathbf{P}$	eigenvector matrix
P2	component of EP (see Figure 17)
P3	EEG scalp location (see Figure 16) or component of EP (see Figure 17) (Note: P3 is used in two very distinct ways in this

dissertation. However, no suitable substitute is appropriate in any either case.)

P4	EEG scalp location (see Figure 16)
PC	principal component matrix
$\mathbf{PC}_i$	principal component for $i = 1,...,I$
Pz	EEG scalp location (see Figure 15)
r	index for range bins
r_s	Spearman rank correlation coefficient statistic
$r_{\alpha,I}$	critical value of Spearman rank correlation test for level of significance α and I feature rankings
R^I	total number of range bins in I-dimensional input feature space
$RMSE_{test}$	test set root mean squared error
$RMSE_{train}$	training set root mean squared error
S_d	sample standard deviation of d
S_i	sample standard deviation of feature $i = 1,...,I$
S_{D_i}	sample standard deviation of $D_{i,g}$
$S_{\hat{\Lambda}_N}$	sample standard deviation of $\hat{\Lambda}_{N,g}$
S_{τ_N}	sample standard deviation of $\tau_{N,g}$
$S_{x_i,x_{i_0}}$	sample covariance between input x_i for $i = 1,...,I$ and input x_{i_0} for $i_0 = 1,...,I$
S_{train}	sample standard deviation of CA_{train}^g for training session $g = 1,...,G$
$S_{train}^2(17\,\text{features})$	sample variance of CA_{train}^g for training session $g = 1,...,G$ with top 17 ranked features
$S_{train}^2(33\,\text{features})$	sample variance of CA_{train}^g for training session $g = 1,...,G$ with all 33 features
$S_{train}^2(\text{Best}\,\overline{CA}_{valid})$	sample variance of CA_{train}^g for training session $g = 1,...,G$ with combination of top ranked features that produces the best $\overline{CA}_{valid}$
$S(t)$	signal at time t
$\text{sig}(a)$	sigmoid nonlinear transfer function
SSE_{test}	test set sum of squared errors
SSE_{train}	training set sum of squared errors
$SSE_{train,m}$	training set sum of squared errors for exemplar $m = 1,...,M$
SSE_{valid}	validation set sum of squared errors
SNR_i	Signal-to-Noise Ratio saliency measure for feature $i = 1,...,I$
SNR_{rank}	Signal-to-Noise Ratio saliency measure for feature with ranking *rank*

$t_{\frac{\alpha}{2},G-1}$	t-value for level of significance $\frac{\alpha}{2}$ with $G-1$ degrees of freedom (two-sided)
$t_{\alpha,G-1}$	t-value for level of significance α and $G-1$ degrees of freedom (one-sided)
$t_{\frac{\alpha}{M},G-1}$	t-value for family level of significance $\frac{\alpha}{M}$ with $G-1$ degrees of freedom (Bonferroni joint)
t_s	t-test statistic
$t_{s,i}$	t-test statistic for feature
T	total length of time
T5	EEG scalp location (see Figure 15)
T6	EEG scalp location (see Figure 15)
$\tanh(a)$	hyperbolic tangent nonlinear transfer function
$U(a,b)$	Uniform random distribution between a and b
$\mathbf{W}$	weight matrix
$\mathbf{W}^1$	first layer weight matrix
w	index for incommensurate sin waves
w	weight vector in Rosenblatt's perceptron
w_0	synaptic weight associated with bias term in Rosenblatt's perceptron
$w_{0,j}^1$	first layer weight for $j=1,\ldots,J$ connecting input bias node to hidden node y_j
w_i	synaptic weight $i=1,\ldots,I$ in Rosenblatt's perceptron
$w_{i,j}^1$	first layer weight for $i=1,\ldots,I$ and $j=1,\ldots,J$ connecting input node x_i to hidden node y_j
$w_{N,j}^1$	first layer weight for $j=1,\ldots,J$ connecting the injected noise feature node to hidden node y_j
$\mathbf{W}^2$	second layer weight matrix
$w_{0,k}^2$	second layer weight for $k=1,\ldots,K$ connecting hidden bias node to output node z_k
$w_{j,k}^2$	second layer weight for $j=1,\ldots,J$ and $k=1,\ldots,K$ connecting hidden node y_j to output node z_k
$\mathbf{W}^{layer}$	weight matrix for a given layer
$\mathbf{W}^{layer+}$	updated weight matrix for a given layer
$\mathbf{W}^{layer-}$	weight matrix for a given layer from the previous epoch
$\mathbf{W}^{layer--}$	weight matrix for a given layer from two epochs past
x	neuron vector in Rosenblatt's perceptron
x_0	bias node

x_i	feature $i = 1,\ldots,I$, input node $i = 1,\ldots,I$
$\bar{x}_i$	mean of feature $i = 1,\ldots,I$
$x_{i,m}$	value of feature $i = 1,\ldots,I$ for exemplar $m = 1,\ldots,M$
$x'_{i,m}$	normalized value of feature $i = 1,\ldots,I$ for exemplar $m = 1,\ldots,M$
$x'_{i,r}$	normalized value of range bin midpoint for feature $i = 1,\ldots,I$ and range bin $r = 1,\ldots,R^I$
$\mathbf{x}'_r$	normalized midpoint vector for range bin $r = 1,\ldots,R^I$
$\mathbf{X}'$	normalized input feature set
$\tilde{\mathbf{X}}'$	mean corrected normalized input feature set
y	output of Rosenblatt's perceptron
y_j	hidden node $j = 1,\ldots,J$
$y_{j,m}$	value of hidden node $j = 1,\ldots,J$ for exemplar $m = 1,\ldots,M$
$y_j\left(\mathbf{x}'_m, \mathbf{W}\right)$	value of hidden node $j = 1,\ldots,J$ for normalized input vector $\mathbf{x}'_m$ for $m = 1,\ldots,M$ and $\mathbf{W}$
$\dot{y}_j\left(\mathbf{x}'_m, \mathbf{W}\right)$	partial derivative of $y_j\left(\mathbf{x}'_m, \mathbf{W}\right)$
$y_j\left(\mathbf{x}'_r, \mathbf{W}\right)$	value of hidden node $j = 1,\ldots,J$ for normalized range bin midpoint $\mathbf{x}'_r$ for $r = 1,\ldots,R^I$ and $\mathbf{W}$
$\dot{y}_j\left(\mathbf{x}'_r, \mathbf{W}\right)$	partial derivative of $y_j\left(\mathbf{x}'_r, \mathbf{W}\right)$
z_k	class $k = 1,\ldots,K$, output node $k = 1,\ldots,K$
$z_{k,m}$	value of output node $k = 1,\ldots,K$ for exemplar $m = 1,\ldots,M$
$z_k\left(\mathbf{x}'_m, \mathbf{W}\right)$	value of output node $k = 1,\ldots,K$ for normalized input vector $\mathbf{x}'_m$ for $m = 1,\ldots,M$ and $\mathbf{W}$
$\dot{z}_k\left(\mathbf{x}'_m, \mathbf{W}\right)$	partial derivative of $z_k\left(\mathbf{x}'_m, \mathbf{W}\right)$
$z_k\left(\mathbf{x}'_r, \mathbf{W}\right)$	value of output node $k = 1,\ldots,K$ for normalized range bin midpoint $\mathbf{x}'_r$ for $r = 1,\ldots,R^I$ and $\mathbf{W}$
$\dot{z}_k\left(\mathbf{x}'_r, \mathbf{W}\right)$	partial derivative of $z_k\left(\mathbf{x}'_r, \mathbf{W}\right)$
$\mathbf{z}_m$	output vector for exemplar $m = 1,\ldots,M$

Abstract

This dissertation research extends the current knowledge of feature saliency in artificial neural networks (ANN). Selecting a good input feature set is crucial to the success of any ANN model. Many feature saliency measures were developed in the last decade for use in feedforward multilayer perceptron (MLP) ANNs. Feature saliency measures allow for the user to rank order the features based upon the saliency, or relative importance, of the features.

This research contributes significantly to the theory of feature saliency. In addition, the techniques developed as part of this research effort are applied to the real-world Air Force problem of classifying pilot workload in addition to classifying air traffic controller workload. This research resulted in the following significant contributions:

- Development of the *Signal-to-Noise Ratio (SNR) saliency measure* to identify salient features in a feedforward MLP ANN used to classify pilot workload as well as air traffic controller workload.

- Empirical evidence that the SNR saliency measure provides rankings that are statistically consistent with that of a partial derivative-based saliency measure and a weight-based saliency measure.

- Development of the *SNR screening method* to identify and remove nonsalient features while maintaining good classification accuracy in a *feedforward MLP ANN* used to classify pilot workload.

- Further application of the SNR screening method to identify and remove nonsalient features while maintaining good classification accuracy in an *Elman RNN* used to estimate pilot workload.

- Development of a *partial derivative-based spatial-temporal saliency measure* to identify salient features in an Elman RNN via unfolding the layers of the network through time used to classify pilot workload.

- Development of a *screening method* that utilizes the partial derivative-based spatial-temporal saliency measure to identify and remove nonsalient features

while maintaining good classification accuracy in an Elman RNN used to classify pilot workload.

- Development of a methodology to study the *memory capacity* of an Elman RNN that utilizes the partial derivative-based spatial-temporal saliency measure.

In summary, this dissertation develops several new saliency measures in addition to several new screening methods for use in several types of ANNs to classify mental workload. In addition, a technique for determining the memory capacity of an Elman RNN was developed.

FEATURE SALIENCY IN ARTIFICIAL NEURAL NETWORKS

WITH APPLICATION TO MODELING WORKLOAD

1 Introduction

1.1 Overview

This dissertation research extends the current knowledge of feature saliency in artificial neural networks (ANN). Dr. Steven K. Rogers, a leading ANN researcher, summed it up the best when he said:

> *Your classifier is only as good as your features. If you find a good set of features, you will have a good classifier.* [116]

Selecting a good input feature set is crucial to the success of any ANN model.

This dissertation research also provides insight to modeling mental workload and in particular, that of pilots and air traffic controllers. Colonel David W. Milam, a retired United States Air Force (USAF) fighter pilot and test pilot who flew with General Chuck Yeager, summed it up best when he said:

> *If we only knew what was going on in the mind of a pilot while he was flying an airplane, we could build better airplanes.* [89]

The feature saliency techniques developed as part of this research effort are applied to the real-world Air Force problem of classifying pilot workload in addition to classifying air traffic controller workload.

1.2 Literature Review

This dissertation research reviewed the areas of ANNs, feature saliency measures, and classifying mental workload.

1.2.1 Artificial Neural Networks (ANN)

An ANN is an ensemble of interconnected nodes. A weight is associated with each interconnection. The foundation for the concept of ANNs is biological in that the nodes of an ANN represent neurons, the interconnections of an ANN represent axons, and the weights of an ANN represent the conversion of action potentials to a chemical ion [58]. An ANN is capable of classification and function estimation [58]. The ability to classify nonlinearly separable (but multi-hyperplane separable) disjoint regions is the greatest attribute of ANNs [115]. An ANN can generalize well for both classification and function estimation problems despite noise in the data [115].

One of the first *neural like* models for pattern recognition was Rosenblatt's perceptron [119, 120, 121, 122, 123]. Rosenblatt's Perceptron provided the building blocks for the feedforward multilayer perceptron (MLP) ANN which is the most widely used ANN today [160]. A feedforward MLP ANN typically has three layers: an input layer, a hidden layer, and an output layer. The feedforward MLP ANN learns by the backpropagation training algorithm as developed by Rumelhart et al. [128] and Werbos [161].

There are several ANNs that can process temporal data. Both the time delay neural network (TDNN) and the recurrent neural network (RNN) allow for the encoding of time [40, 158]. A TDNN is sometimes referred to as a *simple* temporal ANN. A

TDNN typically has three layers: an input layer, a hidden layer, and an output layer. A TDNN embeds time delays on the inputs in a parallel fashion. The user predefines the number of time delays to use. Taken's Theorem [150, 151] as applied by Lapedes [76, 77, 78] is useful for providing an upper and a lower bound on the number of time delays necessary based upon the fractal dimension of the time series inputs to a TDNN. Unfortunately, there are several shortcomings to TDNNs. The most serious of these shortcomings is that a TDNN can not distinguish between relative temporal position and absolute temporal position in a time series [31].

RNNs do not suffer from the shortcomings of a TDNN. Instead of explicitly representing time by embedding time delays on the inputs, a RNN implicitly represents time via a feedback. Most of the shortcomings of a TDNN stem from its representation of time as additional dimensions in the input feature space. A RNN represents time via context nodes by the effect time has on processing the input features. In other words, a RNN is given *memory*. The hidden nodes are fed back in an Elman RNN, the most commonly used RNN, as shown in Figure 10 [31]. An Elman RNN typically has three layers: an input layer, a hidden layer, and an output layer. The output nodes are fed back in a Jordan RNN as shown in Figure 11 [69]. Like an Elman RNN, a Jordan RNN typically has three layers: an input layer, a hidden layer, and an output layer. Both the hidden nodes and the output nodes are fed back in a Williams and Zipser RNN as shown in Figure 12 [166, 167]. A Williams and Zipser RNN typically has two layers: an input layer and a top layer that contains both the hidden nodes and the output nodes.

1.2.2 Feature Saliency Measures

Saliency measures provide a way to measure the relative usefulness of a feature which can be used to rank order the features. There are many reasons why feature saliency is important in ANNs. Both Foley's Rule [37] and Cover's Theorem [23] provide rules of thumb for the appropriate number of input features. Principal component analysis (PCA), one of the classical feature saliency techniques, is based on the normalized eigenvectors and eigenvalues from the covariance matrix of the input feature set [149]. PCA does not depend upon a trained ANN [149]. Ruck's partial derivative-based saliency measure is based upon the sensitivity of a trained ANN's outputs to its inputs and utilizes the sum of the absolute value of the derivative of the outputs with respect to a specific input [124, 126]. Tarr's weight-based saliency measure is based upon the first layer weights of a trained ANN [152].

Feature screening methods, which typically utilize feature saliency measures, are useful for selecting a parsimonious set of salient features while maintaining good classification accuracy. Feature screening methods strive to remove irrelevant features in addition to redundant features. The Setiono-Liu screening method adds a penalty function to the error term so that the weights emanating from necessary inputs will have large magnitudes and the weights emanating from irrelevant input will drop to 0.0 [132]. The Belue-Bauer screening method adds an injected noise feature to provide a baseline for statistically comparing saliency measures that can be either partial derivative-based or weight-based [11, 12]. The Steppe-Bauer screening method improves upon the Belue-Bauer screening method by providing a more powerful statistical test, which can be a paired t-test or a Bonferroni joint test [136, 137, 138, 139].

1.2.3   Classifying Mental Workload

The issue of pilot workload is important to the USAF because pilot overload or task saturation is decreasing mission effectiveness and, in some extreme cases, causing loss of lives [3]. The ability to monitor a pilot's workload will also have far-reaching results in the research and development of future cockpits. Like flying an airplane, air traffic control has long been regarded as a complex, demanding, and at times task saturating endeavor [15, 177]. There is a considerable amount of data from past studies and past experiments that show consistent changes in human physiological responses that are related to the nature and intensity of mental activity. Research to date has been initially successful and shows promise in using electrophysiological measures to distinguish between certain levels of mental workload. Several peripheral psychophysiological measures that show promise and will be used in this dissertation for classifying mental workload include electro-oculography (EOG), electrocardiography (ECG), respiration gauges, and electroencephalography (EEG). Eye blink rate from EOG has been shown to be a sensitive measure to visual workload [15, 169, 170, 175, 177]. Eye Blink rate typically decreases when visual demands increase [15, 169, 170, 175, 177]. Increased heart rate from ECG is typically associated with increased workload [15, 169, 170, 175, 177]. Heart rate from ECG and respiration rate from respiration gauges increase during periods of increased mental workload such as during take-offs and landings [55, 117, 118]. Also, the variability of the cardiac rhythm in addition to the respiration rhythm decrease with increased task difficulty [15, 169, 170, 175, 177].

Ongoing measurements of EEG shows the most promise of being sensitive to the levels or intensity of mental workload. As early as 1929, Hans Berger, the discoverer of

EEG, asked:

> Will is be possible to demonstrate intellectual processes by means of the EEG? [13: 569].

This research will investigate the use of EEG as a measurement of mental workload. In particular, this dissertation will explore the use of preprocessed EEG features for determining the level of a pilot's intellectual processes while flying an airplane in addition to that of an air traffic controller while controller several aircraft. EEG from as little as six channels up to 128 channels can be collected from electrode sites located on the head using the Workload Assessment Monitor (WAM) [172]. Since the brain is the organ responsible for evaluating sensory information and then making and carrying out decisions based upon that sensory information, ongoing activity as measured by EEG would seem to hold a great deal of potential for measuring mental workload. Dr. Glenn F. Wilson, a leading workload researcher, states in the preface of the special issue of *Biological Psychology* on "EEG in Basic and Applied Settings":

> The EEG can be used to derive a more complete understanding of the workings of the human brain and also can be correlated with human performance to provide insights into cognition. [171: vii]

EEG currently appears to be our best "window to the brain."

The use of electrophysiological measures for classifying types of mental activities and for classifying the levels of these mental activities is only in its infancy. The research has shown success but still has a long way to go. There are many obstacles to overcome in using EEG. There are tradeoffs between assumptions, practicality, and accuracy. There are some challenges faced in using EEG which include nonstationarity of the EEG signal, cross-correlation between EEG channels, consistency day-to-day and

individual-to-individual in the EEG channels, quantifying the EEG signal, and appropriately sampling the EEG.

There are filter and detection schemes available for trying to discover, in the EEG, data relevant to some component of human activity. Previous methods include nonparametric methods, parametric methods, mimetic analysis, matched filtering or template matching, and topographic analysis [100].

ANNs show promise for classifying workload using EEG and peripheral psychophysiological data due to the nonlinearity of data, the generalization capabilities of ANNs, and the classification capabilities of ANNs. In particular, TDNNs and RNNs show promise for classifying mental workload due to the temporal nature of EEG and other psychophysiological measures.

1.3 Feasibility Studies Using Time Delay Neural Networks (TDNN) and Recurrent Neural Networks (RNN) to Classify Mental Workload

Two feasibility studies were conducted to investigate the potential use of TDNNs and RNNs to classify mental workload. Prior to this dissertation, TDNNs and RNNs had never been used to classify EPs, ongoing EEG, or mental workload. The first study investigates the feasibility of using a TDNN to detect EPs in an EEG signal. The second study investigates the feasibility of using an Elman RNN to classify mental workload using ongoing EEG activity in the presence of noise.

7

1.3.1 Feasibility of Using Time Delay Neural Networks (TDNN) to Detect Evoked

Potentials (EP)

Only recently have investigators begun to focus on single evoked potential (EP) responses. If EPs are ever to be used to classify pilot workload or air traffic controller workload, then the ability to detect and then classify single EPs must be possible. The major problem is determining what portion of the EEG signal is evoked by the response to the stimulus and what portion represents the continuation of ongoing background EEG.

An EEG signal is generated at a sampling rate of 50 Hz by summing five incommensurate sin waves based, in part, on actual EEG data. A total of 17 rectangle pulses to represent an EP are randomly placed throughout the EEG signal. Five time series are created such that the SNR between the rectangle pulse and the EEG are different: +27.40 dB, +17.41 dB, +7.40 dB, and -2.59 dB, and -12.60 dB. The time series are divided into four classes as follows:

1. EEG only
2. Slight chance that an EP is present
3. EP is more than likely present
4. EP present

Another EEG signal is generated at a sampling rate of 100 Hz by summing five incommensurate sin waves based, in part, on actual EEG data. A total of 17 generated EPs are randomly placed throughout the EEG signal. Five time series are created such that the SNR between the EP and the EEG are different: -1.59 dB, -3.53 dB, -6.02 dB, -9.55 dB, -15.57 dB. As before, the time series are divided into four classes as follows:

1. EEG only
2. Slight chance that an EP is present
3. EP is more than likely present
4. EP present.

A TDNN is trained for each of the varying SNRs via instantaneous backpropagation using a fixed learning rate $\eta = 0.3$ and no momentum. A such, a total of 10 TDNNs are trained. The trained TDNNs for rectangle pulse detection performed adequately when the SNR was +27.40 dB, +17.41 dB, or 7.40 dB. When the SNR was -2.59 dB or -12.60 dB, the TDNN for rectangle pulse detection did not perform adequately. The TDNNs for EP detection performed adequately when the SNR was -1.59 dB, -3.53 dB, -6.02 dB, and -9.55 dB. The TDNN did surprisingly well when the SNR was -9.55 dB. The TDNN for EP detection at -15.57 dB did not perform adequately but it performed better than the TDNN for rectangle pulse detection at -12.82 dB. Since the actual SNR between an EP and EEG is -20 dB, it is clear from this feasibility study that the modeling of pilot workload in addition to air traffic controller workload should not utilize single event EPs.

1.3.2 Feasibility of Using Elman Recurrent Neural Networks (RNN) to Classify Mental Workload Using Ongoing Electroencephalography (EEG)

If an Elman RNN is ever to classify pilot workload using EEG collected during flight, than an Elman RNN classifier must be robust to the effects of noise. There are many sources of potential noise in a cockpit including vibration, movement, talking on the radios, and G forces. For this feasibility study, EEG is collected from a test subject performing three types of mental activity:

1. Reading
2. Eyes open
3. Eyes closed.

An Elman RNN is first trained using 10 features derived from the α-band of EEG to classify the type of mental activity being performed. The features represent the average power of the α-band over a 10-second moving window with 50% overlap in addition to the variance of the power of the α-band over a 10-second moving window with 50% overlap from five electrodes. The Elman RNN is trained via backpropagation with momentum and an adaptive learning rate. The initial learning rate η is set to 0.001. Ten test sets with varying levels of added uniformly distributed noise are used to evaluate the Elman RNN's robustness to noise. The varying levels of noise added to the normalized test exemplars are: 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, and 0.50. The measure of effectiveness (MOE) is the test set classification accuracy CA_{test}.

Next, an Elman RNN is trained using 90 features derived from nine frequency bands of the EEG to classify the type of mental activity being performed. The features represent the average power of nine frequency bands over a 10-second moving window with 50% overlap in addition to the variance of the of the power of nine frequency bands over a 10-second moving window with 50% overlap from five electrodes. Again, 10 test sets with varying levels of added uniformly distributed noise are used to evaluate the Elman RNN's robustness to noise. As before, The varying levels of noise added to the normalized test exemplars are: 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, and 0.50. Again, the MOE is CA_{test}.

In both Elman RNNs trained, the training set classification accuracy CA_{train} with no added noise is 100%. With only 10 input features derived from the α-band, the CA_{test} remains greater than 80% so long as the noise added is no larger than 0.15. With all 90 input features, the CA_{test} remains greater than 80% so long as the noise added is no

larger than 0.30. The Elman RNN trained with 90 features appears to be more robust to the effects of added noise. This feasibility study shows that an Elman RNN can adequately classify among three types of mental activity even in the presence of some added noise. The Elman RNN shows promise for classifying pilot workload in addition to air traffic controller workload.

1.4 Significant Contributions

This research contributes significantly to the theory of feature saliency. In addition, the techniques developed as part of this research effort are applied to the real-world Air Force problem of classifying pilot workload in addition to classifying air traffic controller workload.

1.4.1 Development of the Signal-to-Noise Ratio (SNR) Saliency Measure to Classify Workload

A significant contribution of this research is the development of the SNR saliency measure to identify salient features in a feedforward MLP ANN used to classify pilot workload as well as air traffic controller workload. The SNR saliency measure is a new saliency measure. The SNR saliency measure determines the saliency of a feature by comparing the first layer weights of a feature to that of an injected noise feature. The value of the SNR saliency measure should be significantly larger than 0.0 for salient features and very close to 0.0 or less than 0.0 for nonsalient features. The SNR saliency measure is appealing not only because it can be used to rank order the features from most relevant to least relevant, but that it directly compares the saliency of a feature to that of

11

an injected noise feature.

Saliency measures aid in identifying salient psychophysiological features for classifying workload. Input features are derived from EEG, EOG, ECG, and respiration gauges collected from a pilot or air traffic controller while performing his duties. Several saliency measures are computed including partial derivative-based saliency measures, weight-based saliency measures, and the SNR saliency measure. The rankings derived from the SNR saliency measure are compared to that of a partial derivative-based saliency measure and a weight-based saliency measure. Empirical evidence shows that the SNR saliency measure provides rankings that are *statistically consistent* with that of a partial derivative-based saliency measure and a weight-based saliency measure. This result is another significant contribution of this dissertation.

1.4.1.1 Classifying Pilot Workload

Before this research, a methodology for identifying a salient set of psychophysiological features for classifying pilot workload did not exist. In previous studies, psychophysiological features were selected by maximization of the *CA* by trial and error [175]. The pilot workload data set used involved a simulated landing scenario. The pilot "test subject" started off the scenario descending to an airfield in the clouds. At this point, his workload was classified as low. As soon as the pilot test subject broke through the clouds, his workload was classified as high until touchdown. The scenario ended at touchdown. EEG were collected at six scalp location as shown in Figure 16 and preprocessed into nine frequency bands as listed in Table 4. Features representing the average power and the variance of the power over a 10-second moving window with 50%

overlap were computed for each frequency band at each electrode. Peripheral psychophysiological features computed over a 10-second moving window with 50% overlap included number of eye blinks, heart interbeat interval, the slope of the heart rate, and respiration interbreath interval. Correlation analysis was used as an initial screening method on the features. Eighteen features with a sample correlation greater than 0.75 (with workload) were selected. A feedforward MLP ANN with a 19/19/2 architecture was trained using those 18 features and an injected noise feature.

Two types of partial derivative-based saliency measures, four types of weight-based saliency measures, and the SNR saliency measure were utilized to rank order features used for classifying a pilot's workload as low or high. The top four features for the given pilot test subject were average log of the power of the Δ frequency band at electrodes P3, FZ, and C4 (see Figure 16) along with number of eye blinks. All of the saliency measures provided rankings that were similar. The CA_{test} using 18 features for the given test pilot subject was 90%. The CA_{test} using only the top four salient features was 90%. The CA_{test} did not decrease for the given pilot test subject.

This investigation exhibited the usefulness of ongoing EEG for classifying levels of pilot workload. This research showed that features derived from EEG may have more potential for classifying levels of pilot workload than peripheral psychophysiological features. In addition, this investigation showed that the SNR saliency measure appears to provide rankings consistent with that of partial derivative-based and weight-based saliency measures. Most importantly, this research developed a methodology that can select a set of salient features for classifying pilot workload that does not decrease the CA_{test}.

1.4.1.2 Classifying Air Traffic Controller Workload

The methodology that was developed for determining salient features for classifying pilot workload was also applied to air traffic controllers but in more detail. As with classifying pilot workload, a methodology for identifying a salient set of psychophysiological features for classifying air traffic controller workload did not exist prior to this research. The air traffic controller workload data set used involved simulated air traffic control tasks at Los Angeles International Airport using TRACON (Terminal Radar Approach Control), a computer-based air traffic control simulation []. Four levels of workload were determined by the number of aircraft controlled. The low workload condition consisted of controlling six aircraft in 15 minutes. The medium workload condition consisted of controlling 12 aircraft in 15 minutes. The high workload condition consisted of controlling 18 aircraft in 15 minutes. Finally, the overload condition consisted of controlling 15 aircraft in 5 minutes.

EEG were collected at six scalp location as shown in Figure 15 and preprocessed into five frequency bands as listed in Table 3. Features representing the average power over a 10-second moving window with 50% overlap were computed for each frequency band at each electrode. Peripheral psychophysiological features computed over a 10-second moving window with 50% overlap included average power of the EOG, heart interbeat interval, and respiration interbreath interval.

There were four steps. In the first step, 30 feedforward MLP ANNs with a 33 / 68 / 4 architecture were trained using 33 features. Thirty training sessions were conducted in order to invoke the central limit theorem [88]. $\overline{CA}_{valid}$ over 30 training session was 84.71%. A 95% confidence interval for $\overline{CA}_{valid}$ was (82.69%, 86.73%). The

14

minimum CA_{valid} attained was 73.91% and the maximum CA_{valid} was 95.65%. The $\overline{CA_{valid}}$ over 30 training sessions for the overload condition was 99.46%.

In the second step, 30 feedforward MLP ANNs with a 34 / 68 / 4 architecture were trained using 33 features plus an injected noise feature. The average saliency of the 33 features over 30 training sessions was computed using three types of saliency measures: a partial derivative-based saliency measure, a weight-based saliency measure, and the SNR saliency measure. Regardless of saliency measure, the injected noise feature was always the least salient feature. Of the autonomic nervous system features, respiration interbreath interval was, on average, the most salient feature. The heart interbeat interval was, on average, the second most salient autonomic nervous system feature and power of the EOG signal was, on average, the least salient autonomic nervous system feature. For EEG, those features derived from the $\mu\beta$ frequency band appeared, on average, to be the most salient. In fact, three of the top four most salient features were derived from the $\mu\beta$ frequency band for all three saliency measures. Those features derived from the Δ frequency band appeared, on average, to be the least salient.

In the third step, Spearman rank correlation tests [88] concluded that the rankings from the three types of saliency measures were, on average, statistically consistent with 95% confidence.

In the fourth and final step, 30 training sessions were performed for each combination of the top ranked features. The highest $\overline{CA_{valid}}$ corresponded to the feature set combination that contained the top 17 ranked features. $\overline{CA_{valid}}$ over 30 training session was 87.10%. A 95% confidence interval for $\overline{CA_{valid}}$ was (85.16%, 89.04%). The minimum CA_{valid} attained was 76.09% and the maximum CA_{valid} was 97.83%. A $t-$test

15

concluded that the $\overline{CA}_{valid}$ significantly increased after removing 16 nonsalient features with 95% confidence.

The $\overline{CA}_{valid}$ over 30 training sessions for the overload condition was 99.73%. χ^2 tests performed on the rows of the validation set confusion matrix with 33 features and that with the top 17 ranked features concluded that the $\overline{CA}_{valid}$ for the overload condition significantly increased after removing 16 nonsalient features with 95% confidence. The $\overline{CA}_{valid}$ for the other workload conditions were not effected by removing 16 nonsalient features with 95% confidence.

This air traffic controller workload investigation exhibited the usefulness of ongoing EEG in addition to peripheral psychophysiological measures for classifying levels of air traffic controller workload. Of the top 17 features, all three peripheral psychophysiological features were included. Of the EEG frequency bands, features from the $\mu\beta$ frequency band were selected more often than any other frequency band. Selection of EEG features mostly came from the scalp locations of Fz, T5, and T6 (see Figure 15). In addition, this investigation empirically showed that the SNR saliency measure provides rankings that are, on average, statistically consistent with that of partial derivative-based and weight-based saliency measures. Most importantly, this research developed a methodology that can select a set of salient features for classifying air traffic controller workload that does not decrease the CA_{valid}, on average, and may even statistically increase CA_{valid}, on average.

1.4.2   Development of the Signal-to-Noise Ratio (SNR) Screening Method to Modeling Pilot Workload

Another major contribution of this research is the development of the *SNR screening method*, which utilizes the SNR saliency measure, to identify a parsimonious set of salient features for modeling pilot workload. The SNR screening method identifies and removes nonsalient features while maintaining good classification accuracy. The SNR screening method is employed on a feedforward MLP ANN to classify pilot workload. The SNR screening method is also employed on an Elman recurrent neural network (RNN) to estimate pilot workload.

Since there is a strong temporal component to EEG and the peripheral psychophysiological features, the next logical step was to classify pilot workload using a type of ANN that allows for the encoding of time such as the Elman RNN [31] as depicted in Figure 10. First, an Elman RNN was trained using 112 features over a 5^2 full-factorial design of experiments. A 5^2 full-factorial design of experiments was used to examine the effects of two factors: the momentum constant m_c and the number of hidden/context nodes denoted as J. Five levels of each factor was investigated. Feature inputs were derived from psychophysiological recordings including EEG. Next, the SNR screening method was utilized to select the parsimonious set of salient features over the 5^2 full-factorial design of experiments. An Elman RNN was then trained on the parsimonious set of salient features over the 5^2 full-factorial design of experiments. Results showed that SNR screening method can be successfully employed on an Elman RNN to reduce the root mean squared error (RMSE) of the pilot workload test set, on average, by 67%. When an Elman RNN was used to estimate pilot workload while

17

landing an airplane, a root mean squared error (RMSE) of 0.2893 was attained using all available EEG and peripheral measures as inputs to the Elman RNN which was not acceptable. The SNR screening method was applied which reduced the number of features and achieved a RMSE of only 0.0864. This was the first time that the SNR screening method was applied to an Elman RNN.

1.4.3 Development of a Partial Derivative-Based Spatial-Temporal Saliency Measure to Classifying Pilot Workload

Another significant contribution of this research is the development of a *partial derivative-based spatial-temporal saliency measure* to identify salient features in Elman RNNs to classifying pilot workload. The partial derivative-based spatial-temporal saliency measure is computing by *unfolding the layers* of an Elman RNN. Each unfolded layer represents a time lag. Whereas feature saliency measures in feedforward MLP ANNs typically provide a single measurement for each feature, the partial derivative-based spatial-temporal saliency measure provides a vector measure for each feature. Each element in the vector represents the saliency of the input feature for each time lag. The features can be rank ordered by comparing the saliency vectors.

1.4.4 Development of a Spatial-Temporal Screening Method

Another major contribution of this research is the development of a screening method, which utilizes the partial derivative-based spatial-temporal saliency measure, to identify a parsimonious set of salient features for classifying pilot workload in an Elman RNN. The screening method identifies and removes nonsalient features while

maintaining good classification accuracy.

1.4.5   Determining the Memory Capacity of an Elman Recurrent Neural Network (RNN)

The final major contribution of this research is the development of a methodology for determining the *memory capacity* of an Elman RNN. The memory capacity of an Elman RNN is defined in terms of the number of *unfolded layers* containing salient input and context nodes. The technique developed in this research determines how far back an Elman RNN *remembers*. In other words, the technique ascertains how far back in time the input and context nodes effect the current output of an Elman RNN. This methodology extends the theory of the partial derivative-based spatial-temporal saliency measure and the SNR saliency measure in that the partial derivative-based spatial-temporal saliency measure is calculated and then statistically compared to an injected noise feature. This noise feature provides a baseline for determining the time lag at which a feature provides no more information than noise. In essence, this noise-like feature provides a baseline for determining the memory capacity of RNN. Application to a wave amplitude detection problem, a well known nonlinear process, as shown in Figure 69 demonstrates the utility of this methodology to determine the memory capacity of an Elman RNN.

A total of 52 Elman RNNs with a $2+2/2/1$ architecture as shown in Figure 70 were trained using Equation 50 on the wave amplitude detection problem in order to get 30 sufficiently trained Elman RNNs. An Elman RNN is sufficiently trained if its classification accuracy for the training, test, and validation sets are all greater than 90%. Thirty sufficiently trained Elman RNNs are desired so that the Central Limit Theorem

tendencies may be exploited in performing one-sided t – tests. It appears from the results that the specific Elman RNN applied to this problem has a high likelihood of training to a local minimum. In this case, the error backpropagation algorithm converged to a local minimum 42% of the time. Techniques utilizing simulated annealing may correct for the Elman RNN's apparent high probability of local minima.

The partial derivatives for the input feature, the two context nodes, and the injected noise feature were computed up to eight unfolded layers for the 30 sufficiently trained Elman RNNs. Next, one-sided t – tests were performed at a significance level $\alpha = 0.05$ to determine the layer at which the input and context nodes provided no more information than noise. The t – tests involve computing the sample mean and sample standard deviation of the partial derivatives computed. Care must be taken when computing the sample mean and sample standard deviation for the context nodes due to the *flip-flop* reversing nature of the trained weights associated with context nodes. To alleviate this problem, the sample mean and sample standard deviation were computed over the context node resulting in the maximum partial derivative and for the context node resulting in the minimum partial derivative.

From the t – tests, it was concluded that the following inputs, on average, provided more information than noise to the Elman RNN with 95% confidence: $x(t)$, $x(t-1)$, $x(t-2)$, $x(t-3)$, and $x(t-4)$. It was also concluded that the following context nodes, on average, provided more information than noise to the Elman RNN with 95% confidence: $y_j(t-1)$, $y_j(t-2)$, $y_j(t-3)$, $y_j(t-4)$, and $y_j(t-5)$ for $j = 1,2$. The memory capacity of an Elman RNN with an architecture as that in Figure 70 for this wave amplitude problem is four unfolded layers.

This methodology for determining the *memory capacity* of an Elman RNN

provides insight into the theoretical workings of RNNs. It is now possible to calculate

how far back in time, on average, an Elman RNN *remembers* for a given data set, a given

Elman RNN architecture, and a given noise distribution to the extent that it is appropriate

to measure *memory* by the partial derivative-based saliency measure over time.

1.5   Organization of this Dissertation

This document is written in the following fashion. Chapters 2 through 4 provide

literature reviews of several pertinent subjects. Chapter 2 provides a literature review of

ANNs. Chapter 3 provides a literature review of feature saliency measures. Chapter 4

provides a literature review of classifying mental workload. Chapters 5 provides a

summary of feasibility studies conducted on the use of TDNN and RNNs for classifying

mental workload. Chapter 6 though 9 provide original work that contributes significantly

to the theory and application of ANNs, feature saliency measures, and classifying mental

workload. Chapter 6 provides the development of the SNR saliency measure to classify

mental workload via feedforward MLP ANNs. Chapter 6 summarizes a referee reviewed

Artificial Neural Networks in Engineering (ANNIE) Conference paper entitled "A

Preliminary Investigation of Selection of EEG and Psychophysiological Features for

Classifying Pilot Workload" [46] that was selected as the second runner-up for the best

paper with a novel engineering application at the 1996 ANNIE Conference. In addition,

Chapter 6 summarizes a paper submitted to the *International Journal of Smart*

Engineering System Design entitled "Selection of Psychophysiological Features for

Classifying Air Traffic Controller Workload in Neural Networks" [50]. Chapter 7

21

provides the development of the SNR screening method to modeling pilot workload via a feedforward MLP ANN in addition to an Elman RNN. Chapter 7 summarizes a referee reviewed ANNIE conference paper entitled "Estimating Pilot Workload Using Elman Recurrent Neural Networks: A Preliminary Investigation" [47] in addition to a paper accepted for publication in *Neurocomputing* entitled "Feature Screening Using Signal-to-Noise Ratios" [49]. Chapter 8 provides the development of the partial derivative-based spatial-temporal saliency measure and a screening method that uses the partial derivative-based spatial-temporal saliency measure to classify pilot workload via Elman RNNs. Chapter 9 provides the theory of determining the memory capacity of an Elman RNN and the application of the theory to a wave amplitude detection problem. Chapter 9 summarizes a referee reviewed ANNIE conference paper entitled "Determining the Memory Capacity of an Elman Recurrent Neural Network" [48] that was selected as second runner-up for the best paper with a theoretical development in technique at the 1998 ANNIE Conference. Conclusions and recommendations are provided in Chapter 10.

2.1 Introduction

This chapter provides a literature review of ANNs. Topics covered include the biological and historical foundations of ANNs, Rosenblatt's perceptron, feedforward MLP ANNs, the backpropagation training algorithm, and temporal ANNs to include the Elman RNN, the Jordan RNN, and the Williams and Zipser RNN.

2.2 Biological and Historical Foundations of ANNs

The human brain possesses remarkable processing power. It interprets imprecise information from the senses at an incredible rate. Most impressive of all, the brain learns without any explicit instructions to create the internal representations that make our interpretations possible. For years, science has attempted to mimic the vast capabilities of the human brain. However, much is still unknown about how the brain trains itself to process information. One attempt to model the brain's learning process is to model the work of individual neurons. From these simple neuron models, interconnections are generated and weighted between the neuron nodes. This ensemble of interconnected neuron nodes, termed an ANN, is capable of learning, recognizing patterns, and classifying data.

2.2.1 Biological Foundations

The foundation for the concept of ANNs lies in our understanding of how the biological neuron works. In the human brain, a typical neuron in an excited state conveys

information to other nearby connected neurons via action potentials. The receiving neuron collects its incoming action potentials from numerous nearby connected neurons and converts the action potentials to a chemical ion. The chemical ion then either excites or inhibits activity in the receiving neuron. It is thought that learning occurs within the brain when the conversion of action potentials is altered [115].

An ANN contains an ensemble of interconnected nodes that represent neurons. Each connection in an ANN has a weight that represents the conversion of action potentials to a chemical ion.

2.2.2 Historical Foundations

The first historical work of *neural like* models for pattern recognition was accomplished in 1943 by McCulloch and Pitts [87]. McCulloch and Pitts brought forth the idea of modeling individual neurons as threshold elements in two-class linear machines [87]. The work of McCulloch and Pitts emphasized error-free performance and included the idea of adaptivity or learning [87]. McCulloch and Pitts built upon previous work done by Fisher [35] and Highleyman [59]. In 1936, Fisher published his classical linear discriminant function paper [35] which paved the way for the application of linear discriminant functions to pattern recognition. Highleyman posed the problem in 1962 of finding the optimal (minimum risk) linear discriminant and proposed plausible gradient descent procedures to determine a solution from samples [59].

2.3 Rosenblatt's Perceptron

The perceptron as shown in Figure 1 was demonstrated by Rosenblatt in 1957 in his efforts to develop a two-class linear machine that was biologically inspired by the

$$y = w_0 + \sum_{i=1}^{I} x_i \cdot w_i$$
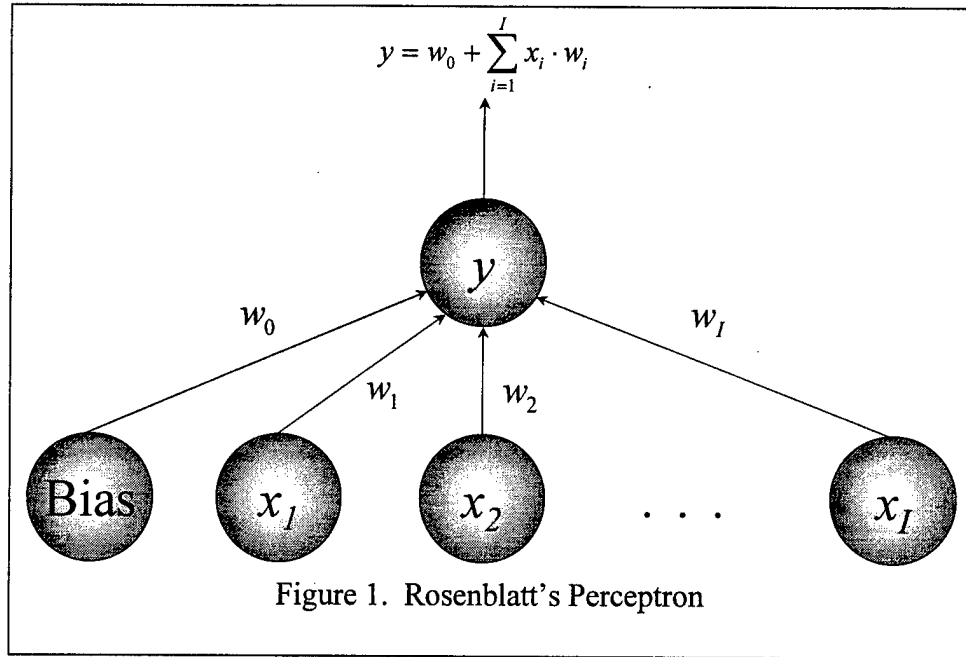
Figure 1. Rosenblatt's Perceptron

human brain [119, 120, 121, 122, 123]. Rosenblatt's work at the time was very controversial but his vision of the human information processing system as a dynamic, interactive, self-organizing system lies at the core of ANNs. The output y of the perceptron is the result of summing each neuron x_i for $i = 1,...,I$ multiplied by its synaptic weight w_i for $i = 1,...,I$ and then adding the synaptic weight associated with the bias term w_0. The bias term allows for the intercept to be non-zero. The surface that separates the two classes can be viewed geometrically as a $(I-1)$ dimensional hyperplane [81] represented by the following equation:

$$\mathbf{x} \cdot \mathbf{w} = 0 \tag{1}$$

If $y < 0$, then the input vector $\mathbf{x}$ is classified as belonging to Class 1. If $y \geq 0$, then the input vector $\mathbf{x}$ is classified as belonging to Class 2. Rosenblatt's perceptron, acting as a pattern classification system, can make limited generalizations and can properly categorize patterns despite noise.

Training a perceptron to classify correctly is equivalent to finding a set of weights **w** such that the hyperplane correctly separates input vectors **x** belonging to Class 1 from those belonging to Class 2. Rosenblatt employed several reinforcement rules for changing the weight vector **w** in his perceptron and hence the orientation and position of the hyperplane in order to improve performance. The best known of these was the fixed increment rule, which guaranteed error free performance whenever it could be achieved for linearly separable two-category problems [119, 120, 121, 122, 123]. Nilsson presented two proofs of the *Perceptron Convergence Theorem* in 1965 for linearly separable two-category problems [103]. The weight vector **w** is updated incrementally during *supervised training* of the perceptron depending upon whether the perceptron classified the input vector **x** correctly. Training is described as supervised when a desired or known output exists for a given input vector **x**. In the case of the perceptron, supervised training occurs when the class of the input vector **x** is known. If the input vector **x** is correctly classified, no adjustments are made to the weight vector **w**. However, if the input vector **x** is incorrectly classified, the old weight vector $\mathbf{w}^-$ is changed to a new weight vector $\mathbf{w}^+$ as follows:

$$\mathbf{w}^+ = \mathbf{w}^- + y \cdot \mathbf{c} \text{ if } y \text{ belongs to Class 1}$$
$$\mathbf{w}^+ = \mathbf{w}^- - y \cdot \mathbf{c} \text{ if } y \text{ belongs to Class 2}$$

$$(2)$$

where **c** is the correction increment vector where each element of **c** is the same constant and can be any fixed number greater than zero.

Unfortunately, the perceptron only saw limited use because it was shown by Minsky and Papert in 1969 to be unable to classify simple Boolean functions like the Exclusive OR (XOR) classification problem as shown in Figure 2 [91, 92]. This was
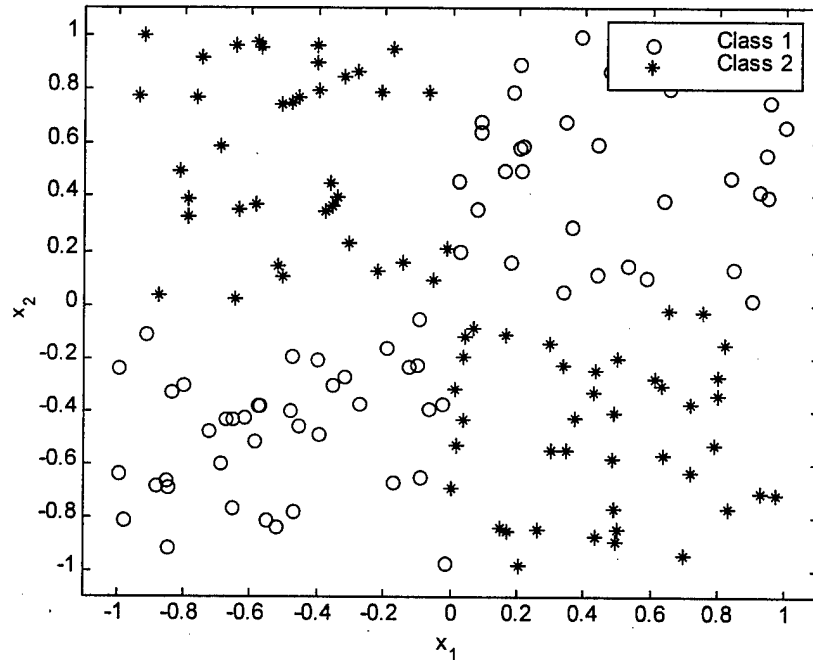
Figure 2. XOR Classification Problem

catastrophic to neural like models. Feedforward MLP ANNs, which we know today easily solve the XOR problem, were in existence at this time but no proven learning algorithms existed for the feedforward MLP ANN [91, 92].

Little research was conducted in neural like models again until 1986 when Rumelhart et al. proposed the backpropagation training algorithm for the feedforward MLP ANN [128]. Though Rumelhart et al. get the credit for developing the backpropagation training algorithm, Werbos was first to derive it in 1974 [161]. Neural like models, in the form of feedforward MLP ANNs, were once again a popular research area.

2.4   Feedforward Multilayer (MLP) Artificial Neural Networks (ANN)

Rosenblatt's perceptron is the basic building block to the feedforward MLP ANN. Feedforward MLP ANNs are necessary to classify the XOR classification problem, more complex classification problems, and nonlinearly separable (but multi-hyperplane separable) classification problems. Feedforward MLP ANNs will even allow for discrimination between disjoint regions. Feedforward MLP ANNs are termed *nonparametric* models because they make no assumptions about the functional form of the underlying population density distribution [160]. In addition, feedforward MLP ANNs make no assumptions about the equality of the covariance matrices between classes.

A feedforward MLP ANN is defined by Hecht-Nielson in the following manner:

> A neural network is a parallel, distributed, information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches ("fans out") into as many collateral connections as desired; each carries the same signal - the processing element output signal. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via impinging connections and on values stored in the processing element's local memory [58: 2-3].

2.4.1   Architecture

Each connection in a feedforward MLP ANN has a weight. The architecture of a feedforward MLP ANN is depicted in Figure 3. A feedforward MLP ANN typically has three layers:

28

- One input layer containing $i = 1,...,I$ input nodes x_i and a bias node x_0.
- One hidden layer containing $j = 1,...,J$ hidden nodes y_j and a bias node y_0.
- One output layer containing $k = 1,...,K$ output nodes z_k.

A *node* (Hecht-Nielson's *element)* in a feedforward MLP ANN is very similar to Rosenblatt's perceptron. The feedforward MLP ANN as shown in Figure 3 has a $I / J / K$ architecture meaning it has I input nodes on the input layer, J hidden nodes on the hidden layer, and K output nodes on the output layer. In the past, feedforward MLP ANNs may have been constructed with more than one hidden layer. Hornik et al. showed in 1989 that only one hidden layer is required to approximate any response surface so long as it contains an adequate number of hidden nodes [62: 360]. The MLP ANN in Figure 3 is a *feedforward* ANN because inputs from the input layer are fed forward to the hidden layer and then fed forward to the output layer. Each input node x_i and the input bias node x_0 is connected to each hidden node y_j via the first layer weight matrix $\mathbf{W}^1$.
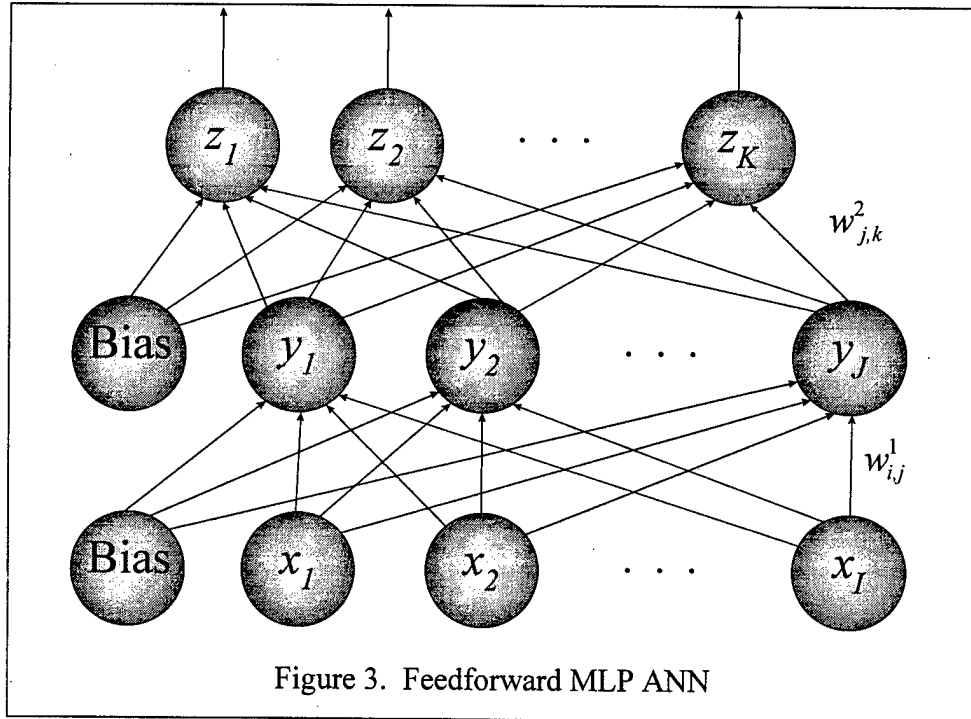
Figure 3. Feedforward MLP ANN

29

Each first layer weight $w_{i,j}^1$ connects input node x_i to hidden node y_j. Similarly, each hidden node y_j and the hidden bias node y_0 is connected to each output node z_k via the second layer weight matrix $\mathbf{W}^2$. Each second layer weight $w_{j,k}^2$ connects hidden node y_j to output node z_k. The feedforward MLP ANN as depicted in Figure 3 is for classification with K output classes. A feedforward MLP ANN can be used for either function estimation or classification.

Figure 4 shows hidden node y_2 in detail. Though Figure 4 is very similar to Rosenblatt's perceptron in Figure 1, there is one major difference: y_2 is activated by a transfer function $f(a)$ where a is the activation into y_2 and is defined as:

$$a = w_{0,2}^1 + \sum_{i=1}^{I} x_i \cdot w_{i,2}^1 \tag{3}$$

2.4.2 Transfer Functions

Many transfer functions exist. The most popular transfer function is the sigmoid

Figure 4. Hidden Node in a Feedforward MLP ANN
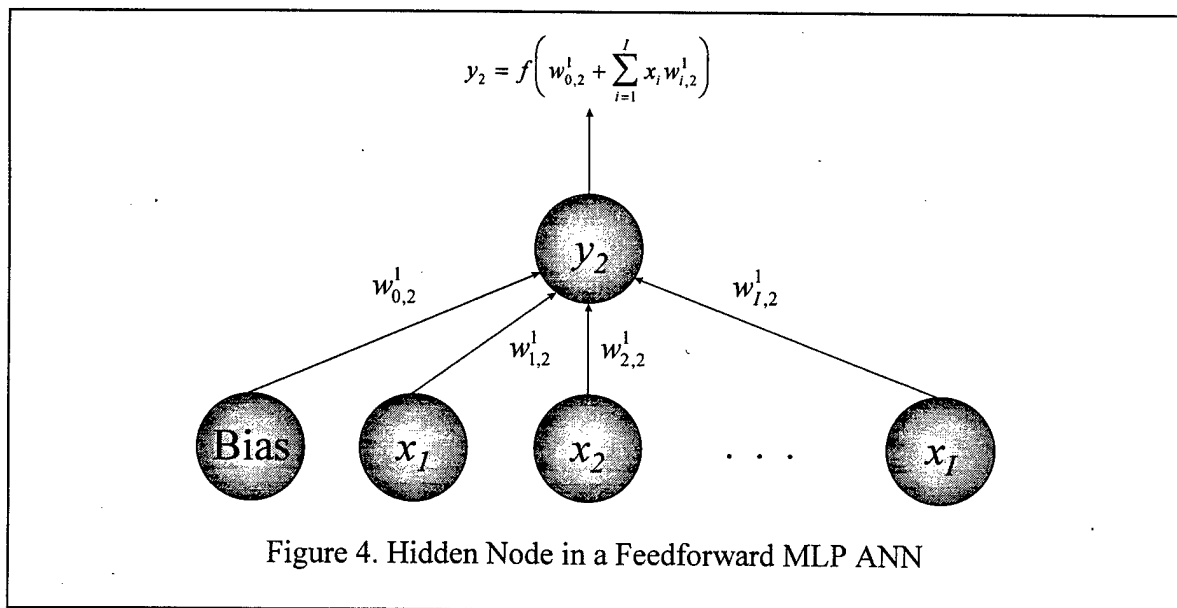
nonlinear transform function:

$$f(a) = \text{sig}(a) = \frac{1}{1 + e^{-a}} \tag{4}$$

which is depicted in Figure 5. Note that $0 < f(a) < 1$ for the sigmoid nonlinear transfer function. The popularity of the sigmoid nonlinear transfer function is due to two reasons. The first is its similarity to the sigmoidal relationship between the excitation and the frequency of firing observed in biological systems. The second is the ease of the

Figure 5. Transfer Functions
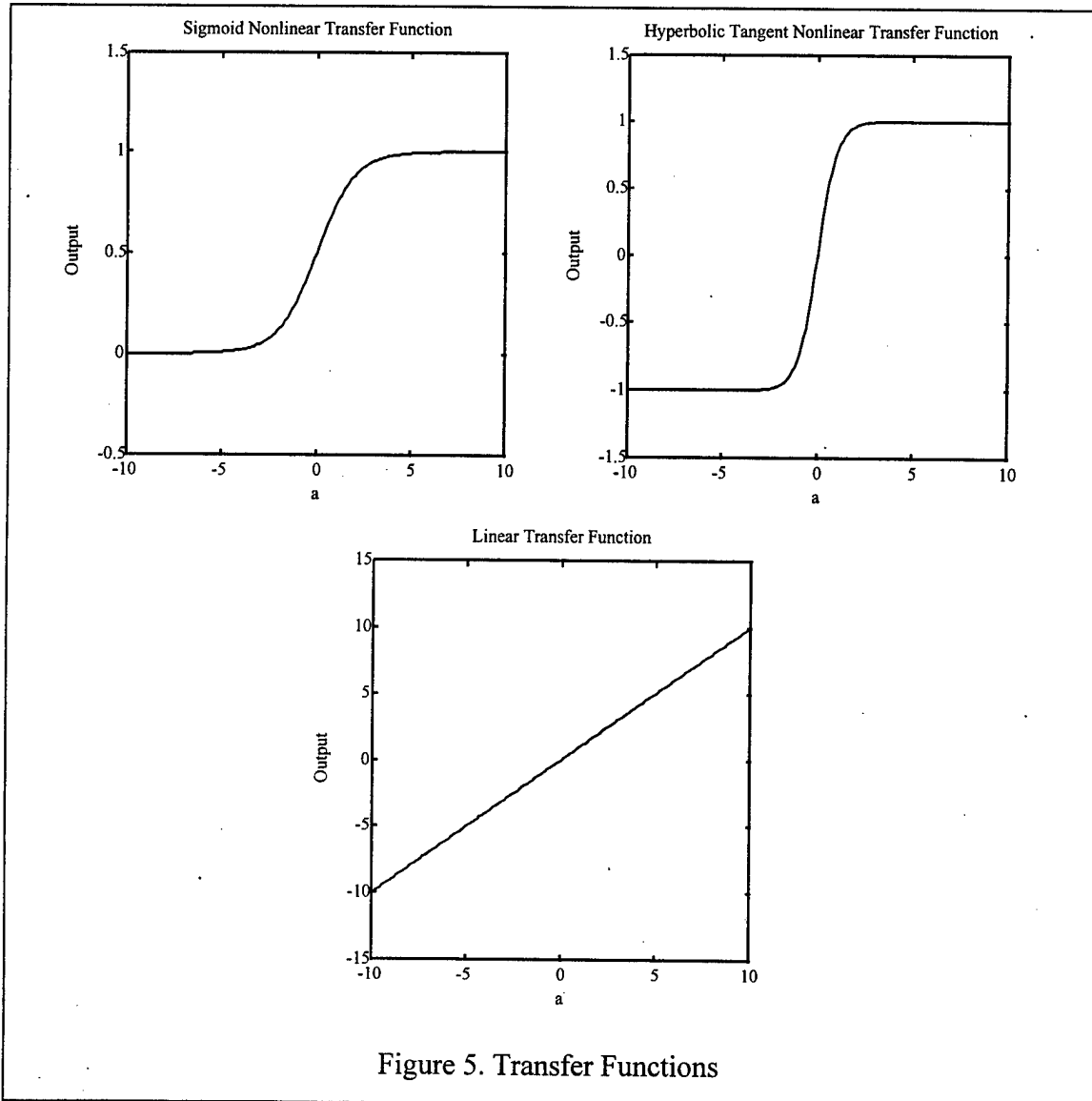
calculation of the first derivative. The first derivative of sigmoid nonlinear transfer function is calculated using the following:

$$\dot{f}(a) = f(a) \cdot [1 - f(a)] \qquad (5)$$

Another popular nonlinear transfer function is the hyperbolic tangent nonlinear transfer function:

$$f(a) = \tanh(a)$$

$$= \frac{\sinh(a)}{\cosh(a)}$$

$$= \frac{\dfrac{e^{-a} - e^{a}}{2}}{\dfrac{e^{a} + e^{-a}}{2}}$$

$$f(a) = \frac{e^{-a} - e^{a}}{e^{a} + e^{-a}} \qquad (6)$$

which is also depicted in Figure 5. Note that $-1 < f(a) < 1$ for the hyperbolic tangent nonlinear transfer function. The first derivative of hyperbolic tangent nonlinear transfer function is calculated using the following:

$$\dot{f}(a) = 1 - [f(a)]^{2} \qquad (7)$$

Also popular and quite simple is the linear transfer function with $\text{slope} = 1$:

$$f(a) = \text{lin}(a) = a \qquad (8)$$

which is also depicted in Figure 5. Note that $-\infty < f(a) < +\infty$ for the linear transfer function with $\text{slope} = 1$. The first derivative of linear transfer function with $\text{slope} = 1$ is easy to calculate using the following:

$$\dot{f}(a) = 1 \qquad (9)$$

32

Rosenblatt's perceptron can be thought of as using the linear transfer function with slope $= 1$. The three more popular transfer functions and their first derivatives are summarized in Table 1.

2.4.3 Backpropagation Training Algorithm

Backpropagation is a gradient descent method that incrementally trains the weights of a MLP ANN via supervised training. Ruck et al. proved in 1990 that the feedforward MLP ANN, when trained as a classifier using backpropagation, approximates by minimum mean squared error (MSE) the Bayes optimal discriminant function for both the two-class problem and the multi-class problem [124, 127]. Ruck et al. showed that this is true regardless of the number of layers and the type of transfer function [124, 127]. Further, outputs of the feedforward MLP ANN approximate the *a posteriori* probability functions of the classes being trained [124, 127].

Since backpropagation is a learning algorithm for supervised training, a desired output vector $\mathbf{d}_m$ exists for every training exemplar $\mathbf{x}_m$ for $m = 1, \dots, M_{train}$. For each layer, the weight matrices $\mathbf{W}^{layer}$ are updated based on backpropagation of the training set sum squared error $SSE_{train,m}$ generated by squaring the difference between the actual output vector $\mathbf{z}_m$ and the desired output vector $\mathbf{d}_m$.

Table 1. Three Popular Transfer Functions and First Derivatives

	Transfer Function $f(a)$	First Derivative $\dot{f}(a)$
Sigmoid	$f(a) = \dfrac{1}{1+e^{-a}}$	$\dot{f}(a) = f(a) \cdot [1 - f(a)]$
Hyperbolic Tangent	$f(a) = \dfrac{e^{-a} - e^{a}}{e^{a} + e^{-a}}$	$\dot{f}(a) = 1 - [f(a)]^2$
Linear	$f(a) = a$	$\dot{f}(a) = 1$

33

2.4.4 Training, Test, and Validation Sets

For supervised training, there is typically a training set and a test set [58]. In some situations, a validation set may also exist [58]. The training set contains M_{train} exemplars, the test set contains M_{test} exemplars, and the validation set contains M_{valid} exemplars. Therefore, the total number of exemplars M is:

$$M = M_{train} + M_{test} + M_{valid} \qquad (10)$$

For feedforward MLP ANNs, exemplars are typically randomly chosen for the training set, test set, and validation set. A good rule of thumb is to randomly select 50% of all available exemplars for the training set, 25% of all available exemplars for the test set, and 25% of all available exemplars for the validation set [115]. In some cases, it may be necessary to randomly select exemplars per class for the training, test, and validation sets.

2.4.5 Measures of Effectiveness

Classification accuracy of the training, test, and validation sets (CA_{train}, CA_{test}, and CA_{valid}) provide excellent measures of effectiveness (MOE) for ANNs used for classification. Various measures of CA include the observed CA, CA confidence intervals, minimum CA, and maximum CA. Confusion matrices also contain valuable information for measuring the effectiveness of ANNs.

2.4.5.1 Observed Classification Accuracy

The observed training set classification accuracy CA_{train} is defined as:

34

$$CA_{train} = \frac{\text{Number of Training Exemplars Classified Correctly}}{M_{train}} \qquad (11)$$

Each exemplar m may be classified by the *winner takes all* scheme on the output nodes such that:

$$Class_m = \begin{cases} 1 \text{ if } \max(z_{k,m}) = z_{1,m} \\ 2 \text{ if } \max(z_{k,m}) = z_{2,m} \\ \quad \vdots \\ K \text{ if } \max(z_{k,m}) = z_{K,m} \end{cases} \quad \text{for } k = 1,2,\ldots K \qquad (12)$$

CA_{test} and CA_{valid} are calculated in a similar fashion to CA_{train} in Equation 11. If $g = 1,2,\ldots,G$ ANNs are trained, then the observed classification accuracy for the gth ANN is denoted CA_{train}^g, CA_{test}^g, and CA_{valid}^g

In addition to the classification accuracy, confusion matrices are helpful to distinguish between the classification accuracy for the K different classes. A confusion matrix such as the one in Table 2 shows the true classification versus the ANN's classification for each of $K = 4$ classes. The diagonal of the confusion matrix contains the number of correct ANN classifications and the CA for each class. The overall CA is

Table 2. Example Confusion Matrix

		Network Classification				
		Class 1	Class 2	Class 3	Class 4	Overall
True	**Class 1**	702 98.46%	1 0.14%	10 1.40%	0 0.00%	713
	Class 2	7 1.00%	657 94.26%	33 4.73%	0 0.00%	697
Classification	**Class 3**	4 0.59%	21 3.11%	650 96.30%	0 0.00%	675
	Class 4	0 0.00%	0 0.00%	0 0.00%	675 100.00%	675
	Overall	713	679	693	675	2760 97.25%

35

found in the bottom right-hand corner. The off-diagonals are the number and percent of classification errors for each type of misclassification. A confusion matrix also contains summary totals in the bottom row and the last column.

2.4.5.2 Confidence Intervals

CA only provides a point estimate. If several ANNs are trained, a confidence interval (CI) for *CA* provides information as to the reliability the observed *CA*. *CA* is a random variable because its value is influenced by the weight initialization and the selection of the training, test, and validation sets. It may be desirable to perform $G \geq 30$ training sessions so that the Central Limit Theorem (CLT) can be invoked [88]. The CLT states that a sample mean will have a sampling distribution that is approximately normal if the sample size is large (i.e. ≥ 30) [88]. For G training sessions, the following CI formula is employed on the training set as:

$$\overline{CA}_{train} - t_{\frac{\alpha}{2},G-1} \cdot \left(\frac{S_{train}}{\sqrt{G}} \right) < \mu_{CA_{train}} < \overline{CA}_{train} + t_{\frac{\alpha}{2},G-1} \cdot \left(\frac{S_{train}}{\sqrt{G}} \right) \tag{13}$$

where $\overline{CA}_{train}$ is the average observed classification for the training set, $t_{\frac{\alpha}{2},G-1}$ comes from any *t*-distribution table, S_{train} is the sample standard deviation of CA_{train}^g for $g = 1,\ldots,G$, and $\mu_{CA_{train}}$ is the expected training set classification accuracy [88]. $\overline{CA}_{train}$ over G training sessions is defined as the following:

$$\overline{CA}_{train} = \frac{\sum_{g=1}^{G} CA_{train}^g}{G} \tag{14}$$

S_{train} is calculated over G training sessions so that:

$$S_{train} = \sqrt{\frac{\sum\limits_{g=1}^{G} \left(CA_{train}^g - \overline{CA}_{train} \right)^2}{G-1}} \qquad (15)$$

All calculations for the test and validation sets are calculated in a fashion similar to Equations 13 - 15.

2.4.5.3 Minimums and Maximums

In addition to reporting $\overline{CA}$ and CIs for μ_{CA}, it may be desirable to compute the minimum CA and the maximum CA attained for the training, test, and validation sets of several ANNs are trained.

2.4.6 Feature Preprocessing

Before the training algorithm can begin, each input feature is typically preprocessed so that the features are "unitless" thus preventing the input features with larger value from dominating. Preprocessing can be done by either standardization or by normalization. The parameters used to preprocess the input features are calculated from the training and test sets. A feature can be standardized via the following:

$$x'_{i,m} = \frac{x_{i,m} - \overline{x}_i}{S_i} \qquad (16)$$

where x'_i is the preprocessed input feature x_i, $\overline{x}_i$ is the sample mean of input feature x_i in the training and test sets, and S_i is the sample standard deviation of input feature x_i in the training and test sets. The sample mean $\overline{x}_i$ is computed as:

$$\bar{x}_i = \frac{\sum_{m=1}^{M_{train,test}} x_{i,m}}{M_{train,test}} \tag{17}$$

The sample standard deviation S_i is computed as:

$$S_i = \sqrt{\frac{\sum_{m=1}^{M_{train,test}} \left(x_{i,m} - \bar{x}_i \right)^2}{M_{train,test} - 1}} \tag{18}$$

After standardization preprocessing, each input feature will have zero mean and unit variance. Depending upon the application, it may be necessary to standardize the training set and the test set individually by replacing $M_{train,test}$ with M_{train} when standardizing the training set and by replacing $M_{train,test}$ with M_{test} when standardizing the test set in Equations 16 through 18 above.

An input feature is typically normalized via the following:

$$x_i' = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{19}$$

where x_i' is preprocessed input feature x_i, $\min(x_i)$ is the minimum value of x_i in the training and test sets, and $\max(x_i)$ is the maximum value of x_i in the training and test sets. After normalization preprocessing in this fashion, each input feature will have values between 0.0 and 1.0. Depending upon the application, it may be necessary to normalize the training set and the test set individually so that $\min(x_i)$ is the minimum value of x_i in the training set and $\max(x_i)$ is the maximum value of x_i in the training set when normalizing the training set using Equation 19. Likewise, $\min(x_i)$ is the minimum value of x_i in the test set and $\max(x_i)$ is the maximum value of x_i in the test set when normalizing the test set using Equation 19.

On occasion, it may be desired to normalize the input features to have values between -1.0 and 1.0 via the following:

$$x_i' = 2 \cdot \left[\frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \right] - 1 \qquad (20)$$

2.4.7 Weight Initialization

The weights must be initialized before training can begin. Weights can be initialized randomly from a Uniform distribution [115]. The range of the Uniform distribution can be as wide as (-0.5, 0.5) or as narrow as (-0.001, 0.001). A more optimal strategy for initializing weights as developed by Nguyen and Widrow decreases training time by more than an order of magnitude [102]. Nguyen and Widrow's weight initialization scheme assumes that each hidden node is responsible for approximating a small portion of the desired output vector **d** and as such, selects initial weights so that each hidden node provides a piece-wise linear approximation to **d** [102].

2.4.8 Mathematics for Weight Updates of Instantaneous Backpropagation

Backpropagation requires that the partial derivative of the sum squared error of the training set denoted as SSE_{train} be computed with respect to each weight in each layer's weight matrix $\mathbf{W}^{layer}$. Define the instantaneous sum squared error $SSE_{train,m}$ associated with exemplar $\mathbf{x}_m$ from the training set as:

$$SSE_{train,m} = \frac{1}{2} \cdot \sum_{k=1}^{K} \left(d_{k,m} - z_{k,m} \right)^2 \qquad (21)$$

where $d_{k,m}$ is the desired output for output node k for exemplar $\mathbf{x}_m$ from the training set

and $z_{k,m}$ is the actual output for output node k for exemplar $\mathbf{x}_m$ from the training set. Note that $SSE_{train,m}$ is a function of the training set and the weights. Note that $SSE_{train,m}$ may be replaced with the mean squared error associated with exemplar $\mathbf{x}_m$ from the training set denoted as $MSE_{train,m}$ and defined as:

$$MSE_{train,m} = \frac{SSE_{train,m}}{K} \qquad (22)$$

The results from the instantaneous backpropagation algorithm using $MSE_{train,m}$ instead of $SSE_{train,m}$ do not differ.

For a given layer in the feedforward MLP ANN, the instantaneous backpropagation learning rule for updating a layer's old weight matrix $\mathbf{W}^{layer-}$ to a layer's new weight matrix $\mathbf{W}^{layer+}$ is the following:

$$\mathbf{W}^{layer+} = \mathbf{W}^{layer-} - \eta \cdot \frac{\partial SSE_{train,m}}{\partial \mathbf{W}^{layer}} \qquad (23)$$

where η is the learning rate and typically $0.0 < \eta < 1.0$ and $\dfrac{\partial SSE_{train,m}}{\partial \mathbf{W}^{layer}}$ is a matrix whose elements are given as:

$$\left(\frac{\partial SSE_{train,m}}{\partial \mathbf{W}^1} \right)_{i,j} = \frac{\partial SSE_{train,m}}{\partial w^1_{i,j}} \text{ for first layer weights}$$

$$\left(\frac{\partial SSE_{train,m}}{\partial \mathbf{W}^2} \right)_{j,k} = \frac{\partial SSE_{train,m}}{\partial w^2_{j,k}} \text{ for second layer weight}$$

For a specific second layer weight $w^2_{j_0,k_0}$ the instantaneous backpropagation learning rule for exemplar $\mathbf{x}_m$ in the training set is:

$$w^{2+}_{j_0,k_0} = w^{2-}_{j_0,k_0} - \eta \cdot \frac{\partial SSE_{train,m}}{\partial w^2_{j_0,k_0}} \qquad (24)$$

40

where the partial derivative of the training set sum squared error $SSE_{train,m}$ with respect to the weight w_{j_0,k_0}^2 is:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = \frac{\partial}{\partial w_{j_0,k_0}^2}\left[\frac{1}{2}\cdot\sum_{k=1}^{K}\left(d_{k,m}-z_{k,m}\right)^2\right] \quad (25)$$

In the partial derivative of the summation in Equation 25, the summation's dependency on w_{j_0,k_0}^2 must be isolated:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = \frac{\partial}{\partial w_{j_0,k_0}^2}\left\{\frac{1}{2}\cdot\left[\begin{array}{l}\left(d_{1,m}-z_{1,m}\right)^2+\left(d_{2,m}-z_{2,m}\right)^2+\cdots+\left(d_{k_0,m}-z_{k_0,m}\right)^2\\ +\cdots+\left(d_{K-1,m}-z_{K-1,m}\right)^2+\left(d_{K,m}-z_{K,m}\right)^2\end{array}\right]\right\} \quad (26)$$

Taking the partial derivative of Equation 26, the only part to survive the differentiation will be variables that involve both subscripts j_0 and k_0. This reasoning identifies the terms $d_{k_0,m}$ and $z_{k_0,m}$. The partial derivative of the summation simplifies to:

$$\frac{\partial\left(d_{k,m}-z_{k,m}\right)^2}{\partial w_{j_0,k_0}^2} = \begin{cases} 0 & \text{if } k \neq k_0 \\ 2\cdot\left(d_{k_0,m}-z_{k_0,m}\right)\cdot\dfrac{\partial\left(-z_{k_0,m}\right)}{\partial w_{j_0,k_0}^2} & \text{if } k = k_0 \end{cases} \quad (27)$$

And thus:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = \left\{\frac{1}{2}\cdot\left[0+0+\cdots+2\cdot\left(d_{k_0,m}-z_{k_0,m}\right)\cdot\frac{\partial\left(-z_{k_0,m}\right)}{\partial w_{j_0,k_0}^2}+\cdots+0+0\right]\right\}$$

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = \left(d_{k_0,m}-z_{k_0,m}\right)\cdot\frac{\partial\left(-z_{k_0,m}\right)}{\partial w_{j_0,k_0}^2} \quad (28)$$

While the desired output $d_{k_0,m}$ is a constant, the actual output $z_{k_0,m}$ is a function of the summation of the weighted outputs from the hidden layer:

$$z_{k_0,m} = f_{k_0}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \qquad (29)$$

where $f_{k_0}(a)$ is the transfer function at output node k_0. The partial derivative of $z_{k_0,m}$

can be written as:

$$\frac{\partial\left(z_{k_0,m}\right)}{\partial w_{j_0,k_0}^2} = \frac{\partial}{\partial w_{j_0,k_0}^2} f_{k_0}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \qquad (30)$$

And therefore:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = -\left(d_{k_0,m} - z_{k_0,m}\right) \cdot \frac{\partial}{\partial w_{j_0 k_0}^2}\left[f_{k_0}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right)\right]$$

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = -\left(d_{k_0,m} - z_{k_0,m}\right) \cdot \dot{f}_{k_0}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \cdot \frac{\partial}{\partial w_{j_0 k_0}^2}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \quad (31)$$

where $\dot{f}_{k_0}(a)$ represents the derivative of $f_{k_0}(a)$ with respect to a. For clarity, let

$$\dot{z}_{k_0,m} = \dot{f}_{k_0}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \qquad (32)$$

Substituting Equation 32 into Equation 31:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = -\left(d_{k_0,m} - z_{k_0,m}\right) \cdot \dot{z}_{k_0,m} \cdot \frac{\partial}{\partial w_{j_0 k_0}^2}\left(w_{0,k_0}^2 + \sum_{j=1}^{J} w_{j,k_0}^2 \cdot y_{j,m}\right) \qquad (33)$$

where

$\dot{z}_{k_0,m}(t) = z_{k_0,m}(t) \cdot \left[1 - z_{k_0,m}(t)\right]$ for sigmoid nonlinear transfer functions,

$\dot{z}_{k_0,m}(t) = 1 - \left[z_{k_0,m}(t)\right]^2$ for hyperbolic tangent nonlinear transfer functions, and

$\dot{z}_{k_0,m}(t) = 1$ for linear transfer functions with slope $= 1$.

Taking the partial derivative in Equation 33, the only variables to survive the

differentiation will be those that involve the subscript j_0:

$$\frac{\partial SSE_{train,m}}{\partial w_{j_0,k_0}^2} = -\left(d_{k_0,m} - z_{k_0,m}\right) \cdot \dot{z}_{k_0,m} \cdot y_{j_0,m} \tag{34}$$

And thus, for a specific second layer weight w_{j_0,k_0}^2, the instantaneous backpropagation learning rule for exemplar $\mathbf{x}_m'$ in the training set is simplified to:

$$w_{j_0,k_0}^{2+} = w_{j_0,k_0}^{2-} + \eta \cdot \left(d_{k_0,m} - z_{k_0,m}\right) \cdot \dot{z}_{k_0,m} \cdot y_{j_0,m} \tag{35}$$

Similarly to Equation 24, the instantaneous backpropagation learning rule for a specific first layer weight w_{i_0,j_0}^1 for exemplar $\mathbf{x}_m'$ in the training set is:

$$w_{i_0,j_0}^{1+} = w_{i_0,j_0}^{1-} - \eta \cdot \frac{\partial SSE_{train,m}}{\partial w_{i_0,j_0}^1} \tag{36}$$

where the partial derivative of the training set sum squared error $SSE_{train,m}$ with respect to the weight w_{i_0,j_0}^1 is:

$$\frac{\partial SSE_{train,m}}{\partial w_{i_0,j_0}^1} = \frac{\partial}{\partial w_{i_0,j_0}^1}\left[\frac{1}{2} \cdot \sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right)^2\right]$$

$$= \sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right) \cdot \frac{\partial\left(-z_{k,m}\right)}{\partial w_{i_0,j_0}^1}$$

$$= -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right) \cdot \frac{\partial}{\partial w_{i_0,j_0}^1}\left[f_k\left(w_{0,k}^2 + \sum_{j=1}^{J}w_{j,k}^2 \cdot y_{j,m}\right)\right]$$

$$= -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right) \cdot \dot{f}_k\left(w_{0,k}^2 + \sum_{j=1}^{J}w_{j,k}^2 \cdot y_{j,m}\right) \cdot \frac{\partial}{\partial w_{i_0,j_0}^1}\left(w_{0,k}^2 + \sum_{j=1}^{J}w_{j,k}^2 \cdot y_{j,m}\right)$$

$$= -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right) \cdot \dot{z}_{k,m} \cdot \frac{\partial}{\partial w_{i_0,j_0}^1}\left(w_{0,k}^2 + \sum_{j=1}^{J}w_{j,k}^2 \cdot y_{j,m}\right)$$

$$\frac{\partial SSE_{train,m}}{\partial w^1_{i_0,j_0}} = -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right)\cdot \dot{z}_{k,m} \cdot w^2_{j_0,k} \cdot \frac{\partial y_{j_0,m}}{\partial w^1_{i_0,j_0}} \tag{37}$$

The output of a hidden layer node is:

$$y_{j_0,m} = f_{j_0}\left(w^1_{0,j_0} + \sum_{i=1}^{I} w^1_{i,j_0}\cdot x_{i,m}\right) \tag{38}$$

Substituting Equation 38 into Equation 37 gives:

$$\frac{\partial SSE_{train,m}}{\partial w^1_{i_0j_0}} = -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right)\cdot \dot{z}_{k,m} \cdot w^2_{j_0,k} \cdot \frac{\partial}{\partial w^1_{i_0,j_0}}\left[f_{j_0}\left(w^1_{0,j_0} + \sum_{i=1}^{I} w^1_{i,j_0}\cdot x_{i,m}\right)\right]$$

$$\frac{\partial SSE_{train,m}}{\partial w^1_{i_0j_0}} = -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right)\cdot \dot{z}_{k,m} \cdot w^2_{j_0,k} \cdot \dot{f}_{j_0}\left(w^1_{0,j_0} + \sum_{i=1}^{I} w^1_{i,j_0}\cdot x_{i,m}\right)\cdot \frac{\partial}{\partial w^1_{i_0,j_0}}\left(w^1_{0,j_0} + \sum_{i=1}^{I} w^1_{i,j_0}\cdot x_{i,m}\right) \tag{39}$$

For clarity, let

$$\dot{y}_{j_0,m} = \dot{f}_{j}\left(w^1_{0,j} + \sum_{i=1}^{I} w^1_{i,j}\cdot x_{i,m}\right) \tag{40}$$

Substituting Equation 40 into Equation 39:

$$\frac{\partial SSE_{train,m}}{\partial w^1_{i_0j_0}} = -\sum_{k=1}^{K}\left(d_{k,m} - z_{k,m}\right)\cdot \dot{z}_{k,m} \cdot w^2_{j_0,k} \cdot \dot{y}_{j_0,m}\cdot \frac{\partial}{\partial w^1_{i_0,j_0}}\left(w^1_{0,j_0} + \sum_{i=1}^{I} w^1_{i,j_0}\cdot x_{i,m}\right) \tag{41}$$

where

$$\dot{y}_{j_0,m}(t) = y_{j_0,m}(t)\cdot\left[1 - y_{j_0,m}(t)\right]\text{ for sigmoid nonlinear transfer functions,}$$

$$\dot{y}_{j_0,m}(t) = 1 - \left[y_{j_0,m}(t)\right]^2\text{ for hyperbolic tangent nonlinear transfer functions, and}$$

$$\dot{y}_{j_0,m}(t) = 1\text{ for linear transfer functions with slope} = 1.$$

Realizing that

$$\frac{\partial\left(w^1_{i,j_0}\cdot x_{i,m}\right)}{\partial w^1_{i_0,j_0}} = \begin{cases} 0 & \text{if } i \neq i_0 \\ x_{i_0,m} & \text{if } i = i_0 \end{cases} \tag{42}$$

then Equation 41 becomes:

$$\frac{\partial SSE_{train,m}}{\partial w_{i_0,j_0}^1} = -\sum_{k=1}^{K} \left(d_{k,m} - z_{k,m}\right) \cdot \dot{z}_{k,m} \cdot w_{j_0,k}^2 \cdot \dot{y}_{j_0,m} \cdot x_{i_0,m} \qquad (43)$$

Therefore, for a specific first layer weight w_{i_0,j_0}^1 the instantaneous backpropagation learning rule is simplified to:

$$w_{i_0,j_0}^{1+} = w_{i_0,j_0}^{1-} + \eta \cdot \sum_{k=1}^{K} \left(d_{k,m} - z_{k,m}\right) \cdot \dot{z}_{k,m} \cdot w_{j_0,k}^2 \cdot \dot{y}_{j_0,m} \cdot x_{i_0,m} \qquad (44)$$

The above derivations for the weight updates are for *instantaneous* backpropagation. Backpropagation is instantaneous whenever the weights are updated after every exemplar in the training set is presented. In other words, the gradient calculation is based on the error from a single training exemplar. For each training cycle or *epoch*, each exemplar in the training set is inputted to the feedforward MLP ANN in random order. Thus for any epoch during training, the weights are updated M_{train} times. For each epoch, the order of presentation of the exemplars is randomly changed.

2.4.9 Batch Backpropagation

Another variation is *batch* backpropagation where the weights are updated after all of the exemplars in the training set are presented. In other words, the gradient calculation is based on the total sum squared error from all of the training exemplars. Thus for any epoch during training, the weights are updated only once. There is no need to randomly change the order of presentation of the training exemplars during batch backpropagation. The total error SSE_{train} for all exemplars $\mathbf{x}_m$ for $m = 1,2,\ldots,M_{train}$ is defined as the following:

$$SSE_{train} = \sum_{m=1}^{M_{train}} SSE_{train,m} \qquad (45)$$

45

where $SSE_{train,m}$ is computed via Equation 21. Note that like $SSE_{train,m}$, SSE_{train} is a function of the training set and the weights. The backpropagation learning rule for batch training is:

$$\mathbf{W}^{layer^+} = \mathbf{W}^{layer^-} - \eta \cdot \frac{\partial SSE_{train}}{\partial \mathbf{W}^{layer}} \tag{46}$$

Derivations for batch backpropagation for a specific weight are similar to those developed for instantaneous backpropagation. Note that like instantaneous backpropagation, SSE_{train} may be replaced with the mean squared error for all exemplars $\mathbf{x}_m$ from the training set denoted as MSE_{train} and defined as:

$$MSE_{train} = \frac{SSE_{train,m}}{K \cdot M_{train}} \tag{47}$$

The results from the batch backpropagation algorithm using MSE_{train} instead of SSE_{train} do not differ. SSE and MSE can also be calculated for the test and validation sets in the same fashion as Equations 45 and 47. SSE and MSE for the training, test, and validation sets provide excellent MOEs for an ANN. Another form of the error that provides an excellent MOE for an ANN is the root mean squared error (RMSE). The RMSE for the training set denoted as $RMSE_{train}$ is defined as:

$$RMSE_{train} = \sqrt{MSE_{train}} \tag{48}$$

$RMSE$ can also be calculated for the test and validation sets in the same fashion as Equation 48.

2.4.10 Momentum

The convergence of backpropagation is slow. One approach for speeding up the

46

training of a feedforward MLP ANN via the backpropagation learning rule is the *momentum method*. Momentum allows for the learning algorithm to respond not only to the local error gradient but to recent trends in the error surface. With momentum, the backpropagation learning rule becomes:

$$\mathbf{W}^{layer+} = \mathbf{W}^{layer-} - \eta \cdot \frac{\partial SSE_{train}}{\partial \mathbf{W}^{layer-}} + m_c \cdot \left(\mathbf{W}^{layer-} - \mathbf{W}^{layer--} \right) \qquad (49)$$

where m_c is the momentum constant and typically $0.0 < m_c < 1.0$ and $\mathbf{W}^{layer--}$ is a layer's weight matrix one epoch before $\mathbf{W}^{layer-}$. Variations do exist for momentum and include the following *MATLAB* implementation:

$$\mathbf{W}^{layer+} = \mathbf{W}^{layer-} + \left(1 - m_c\right) \cdot \eta \cdot \frac{\partial SSE_{train}}{\partial \mathbf{W}^{layer-}} + m_c \cdot \left(\mathbf{W}^{layer-} - \mathbf{W}^{layer--} \right) \qquad (50)$$

MATLAB implements momentum only if the ratio of the new error to the old error falls below a predefined criterion such as 1.04 [24]. In other words, if SSE_{train} increases more than 4% as a result of a weight update, the weight update is recalculated and $m_c = 0$.

2.4.11 Adaptive Learning Rate

Another approach for speeding up the training is an adaptive learning rate η. If the learning rate is too large, the backpropagation algorithm may continually jump over the minimum and never converge. If the learning rate is too small, though, the training will take a long time to converge. The solution is to use an adaptive learning rate that allows for both large and small steps in the weight updates depending upon the complexity of the error-weight space. With *MATLAB*'s implementation, the learning rate is decreased by multiplying it by 0.7 if SSE_{train} increases more than 4% as a result of a

weight update [24]. If SSE_{train} decreases, the learning rate is increased by multiplying it by 1.05 [24]. White suggests a declining learning rate that is a function of the number of epochs $e = 1,2,...,E$ [162]. Steppe provides three types of declining learning rates that are a function of number of epochs $e = 1,2,...,E$ [136]. The *log declining learning rate* is computed as:

$$\eta(e) = \frac{1}{\ln(1+e)} \tag{51}$$

Note that the learning rate is called a *log* declining learning rate though the natural logarithm (ln) is used in the calculation. The *linearly declining learning rate* is computed as:

$$\eta(e) = \eta_0 \cdot \left(1 - \frac{e}{E+1}\right) \tag{52}$$

where η_0 is the initial learning rate and E is the total number of epochs. The *log-linearly declining learning rate* is computed as:

$$\eta(e) = \frac{1 - \dfrac{e}{E+1}}{\ln(1+e)} \tag{53}$$

As with Equation 51, note that the learning rate is called a *log-linearly* declining learning rate though ln is used in the calculation. Of the three declining learning rates $\eta(e)$, Steppe recommends the log declining learning rate in Equation 51 since E which is necessary for both the linearly declining learning rate and the log-linearly declining learning rate is usually not known in advance [136].

2.4.12 Stopping Criterion

There are many stopping criteria for the backpropagation learning algorithm. The learning algorithm may be stopped after a pre-defined number of epochs. For example, the feedforward MLP ANN may be trained for 2000 epochs. The stopping criteria for the backpropagation algorithm may be a pre-defined error minimum. For example, the feedforward MLP ANN may be trained until the SSE_{train} falls below 10.0. In order to prevent memorization of the training set, however, it is typical to train for a pre-defined number of epochs and then keep the set of weights that produced the minimum SSE_{test} [24]. An example of this is given in Figure 6 where the set of weights from epoch 238 are retained.

2.5 Temporal ANNs

Since biological neural networks can process temporal data, it can be assumed that ANNs which were originally formulated from biological research can do the same. The two general types of temporal ANNs are the time delay neural network (TDNN) and the recurrent neural network (RNN). The three types of RNNs include the Elman RNN [31], the Jordan RNN [69], and the Williams and Zipser RNN [166, 167].

2.5.1 Time Delay Neural Network (TDNN)

A TDNN as shown in Figure 7 is a special type of ANN that allows for the encoding of time. The simplest solution to encoding time is to attempt to *parallelize* time on the input layer via a TDNN. A TDNN embeds time delays on the inputs to encode and learn these temporal sequences. Each connection in a TDNN has a weight. A TDNN

Figure 6. *SSE~train~* and *SSE~test~* During Training
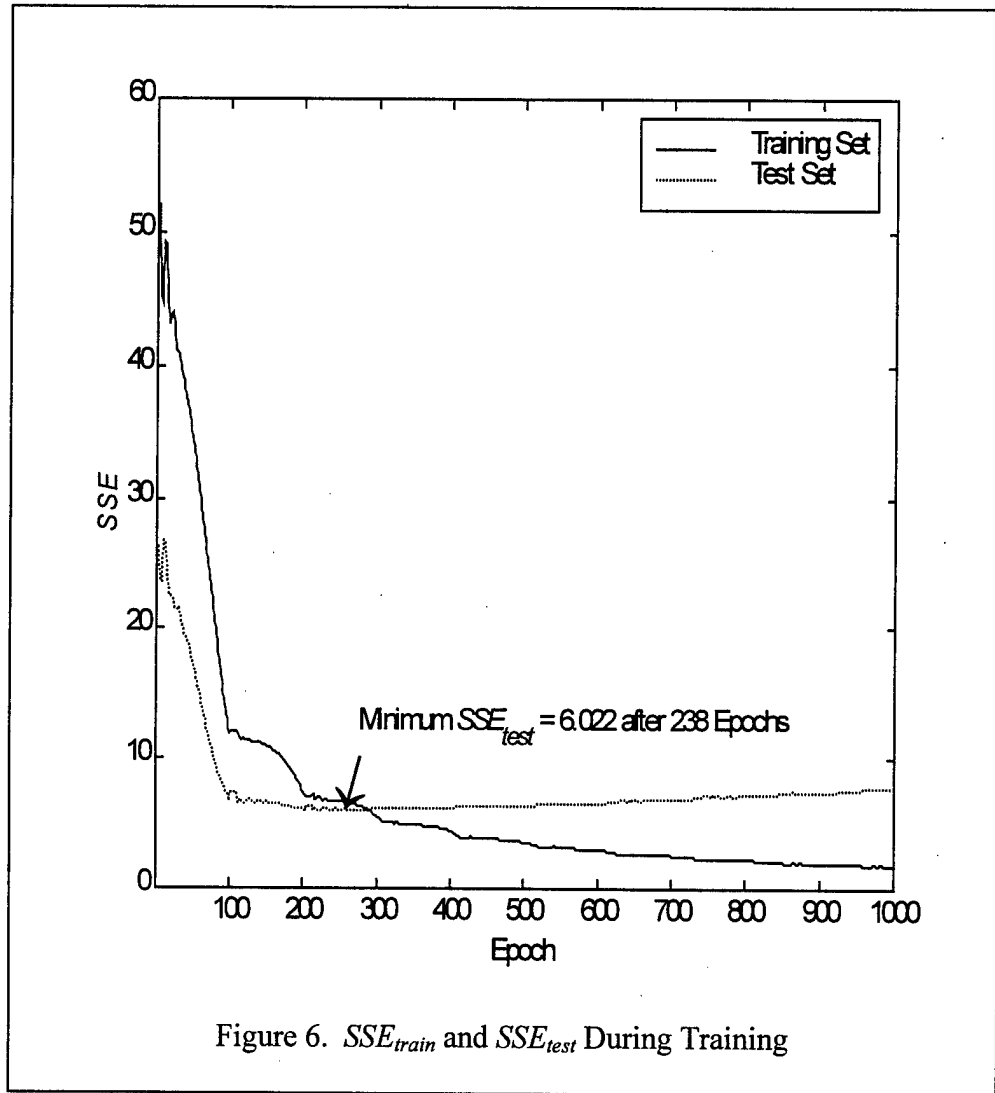
typically has three layers:

- One input layer containing $i = 1, \ldots, I$ input nodes x_i that are each lagged L times and a bias node. For each input $i = 1, \ldots, I$, the lagged inputs are $x_i(t)$, $x_i(t-1)$, $x_i(t-2)$, $\ldots$, $x_i(t-L)$.

- One hidden layer containing $j = 1, \ldots, J$ hidden nodes at time t denoted as $y_j(t)$ and a bias node.

- One output layer containing $k = 1, \ldots, K$ output nodes at time t denoted as $z_k(t)$.

50
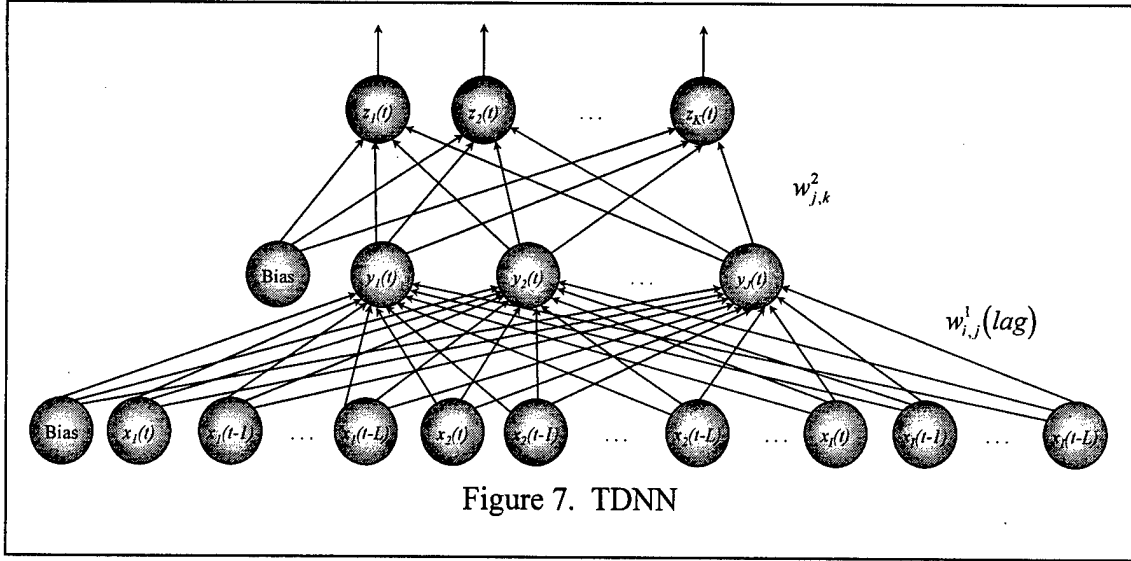
Figure 7. TDNN

The TDNN as shown in Figure 7 has a $I \cdot (L+1)/J/K$ architecture meaning it has

$I \cdot (L+1)$ input nodes on the input layer, J hidden nodes on the hidden layer, and K

output nodes on the output layer. The TDNN as depicted in Figure 7 is for classification

with K output classes. A TDNN can be used for either function estimation or function

prediction in addition to classification or classification prediction. Waibel et al. used a

TDNN to recognize Japanese phonemes [158] and Gainey used a TDNN to predict

British pound opening prices [40].

The user defines the number of inputs, the number of output classes, the number

of time delays, and the number of hidden nodes. The activations on the hidden and

output nodes can be any of the transfer functions as given in Table 1. The output of a

TDNN is based upon the current and lagged inputs. The architecture utilizes a fixed

number of the actual time sequence values as inputs thus spatially presenting a fixed

window of the time sequence. After the architecture of a TDNN is selected, it is trained

via the backpropagation method either instantaneously or by batch. Whereas a

feedforward MLP ANN will have first layer weights $w_{i,j}^1$ for $i = 1,2,\ldots,I$ input features

51

and $j = 1, 2, \ldots, J$ hidden nodes, a TDNN will have first layer weights $w_{i,j}^1(lag)$ for $i = 1, 2, \ldots, I$ input features, $j = 1, 2, \ldots, J$ hidden nodes, and $lag = 0, 1, \ldots, L$ lagged time delays.

2.5.1.1 Use of Fractal Dimension in Time Delay Neural Networks (TDNN)

The fractal dimension of the input time series to a TDNN may be helpful in determining the number of time delays. The fractal dimension provides a measurement of the order and randomness of a time series. Time series with high fractal dimensions are more random than time series with low fractal dimensions. A time series with a low fractal dimension may contain enough order to be predictable [33]. The fractal dimension d_f of an attractor A is defined as

$$d_f(A) = \lim_{\varepsilon \to 0} \left\{ \frac{\log[N(A, \varepsilon)]}{\log(\varepsilon)} \right\} \qquad (54)$$

where $N(A, \varepsilon)$ is the smallest number of squares with side length $\varepsilon > 0$ required to cover A [38]. There is a lack of agreement in the literature for defining an attractor. Milnor's definition states that an attractor of x contained in a metric space is the set of accumulation points for the sequence $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}, \ldots$ [90]. The plots in Figure 8 are provided to illustrate the *box counting* calculation of $d_f(A)$ in Equation 54. A simple two-dimensional attractor with $d_f(A) = 1.0$ is shown. Values of $\varepsilon = 0.5$, 0.25, and 0.1 are used. $N(A, \varepsilon)$ is determined by counting the number of boxes that contain A. Figure 8 shows that as $\varepsilon \to 0$, $d_f(A) \to 1.0$.

52
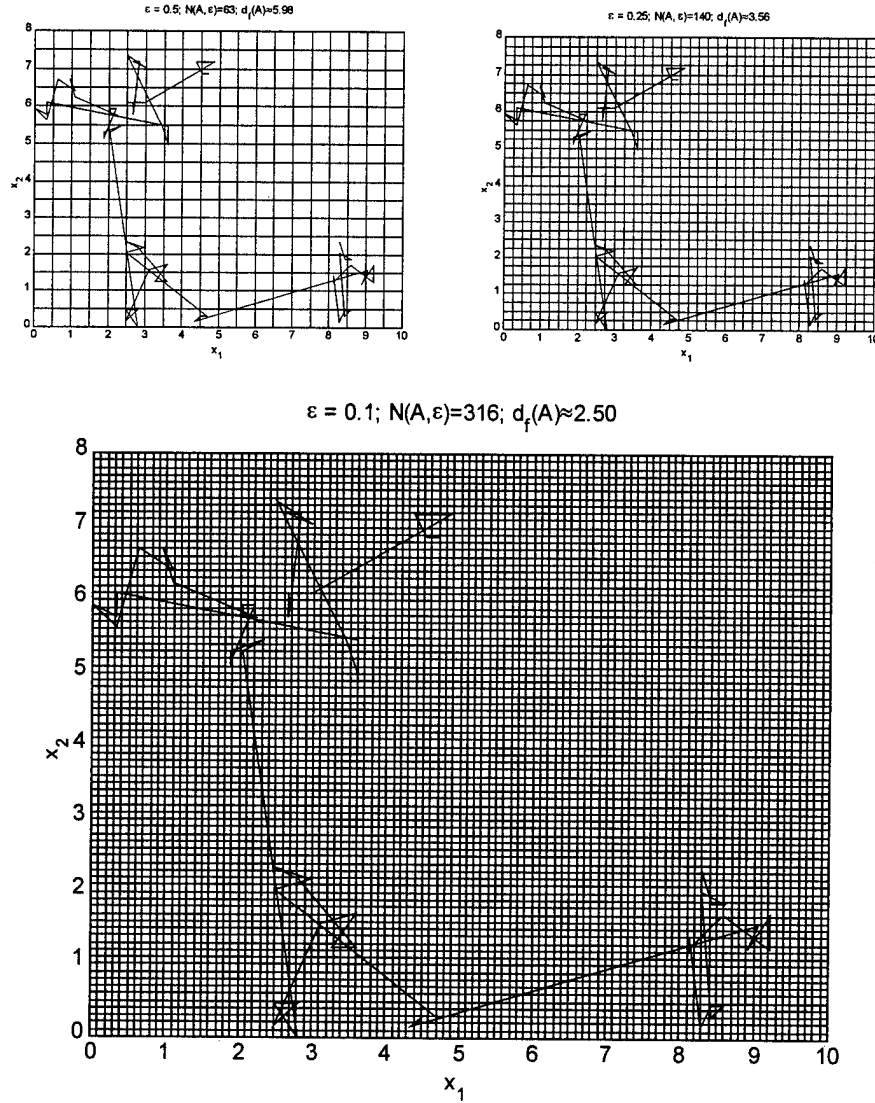
Figure 8. Simple 2-D Attractor with $d_f(A) = 1.0$

Grassberger and Procaccia provide a methodology for estimating the fractal dimension d_f of a time series $\mathbf{x}$ by embedding the time series into a *phase* space and then extracting the fractal dimension of the attractor associated with the time series [45]. This method creates a p-dimensional space consisting of p-tuples taken from the time series [45]. The number of dimensions p chosen is an estimate slightly larger than a estimate of the fractal dimension such that

53

$$d_f(\mathbf{x}) < p < d_f(\mathbf{x}) + 6 \qquad (55)$$

A set of small numbers q_u for $u = 1, 2, \ldots, 6$ is chosen and for each u, the number of pairs

of contiguous p-tuples $\left(x_1, x_2, \ldots, x_p\right)$ within Euclidean distance q_u of each other is

determined and denoted $C(q_u)$ [45]. If $f_d(\mathbf{x}) < 7$ and an adequate number of sample

points are used, the Grassberger and Procaccia method will produce pairs of

$\left(\log(q_u), \log[C(q_u)]\right)$ where $\log(x) = \log_{10}(x)$ so that when plotted are logarithmically

linear and collinear to each other [45]. The average of the slopes of these plotted lines is

a good approximation of the fractal dimension of the time series [45]. Figure 9 provides

an example of the Grassberger and Procaccia method. The time series used in the

example is a sum of two sin waves with incommensurate frequencies. Frequencies are

incommensurate if their ratio is an irrational number. Specifically, the time series used in

Figure 9 is

$$x(t) = \frac{2 + \sin\left(\sqrt{2} \cdot t\right) + \sin\left(\sqrt{3} \cdot t\right)}{2} \qquad (56)$$

Figure 9 shows the pairs of $\left(\log(q_u), \log[C(q_u)]\right)$ using the Grassberger and Procaccia

Figure 9. Example of Grassberger and Procaccia Method
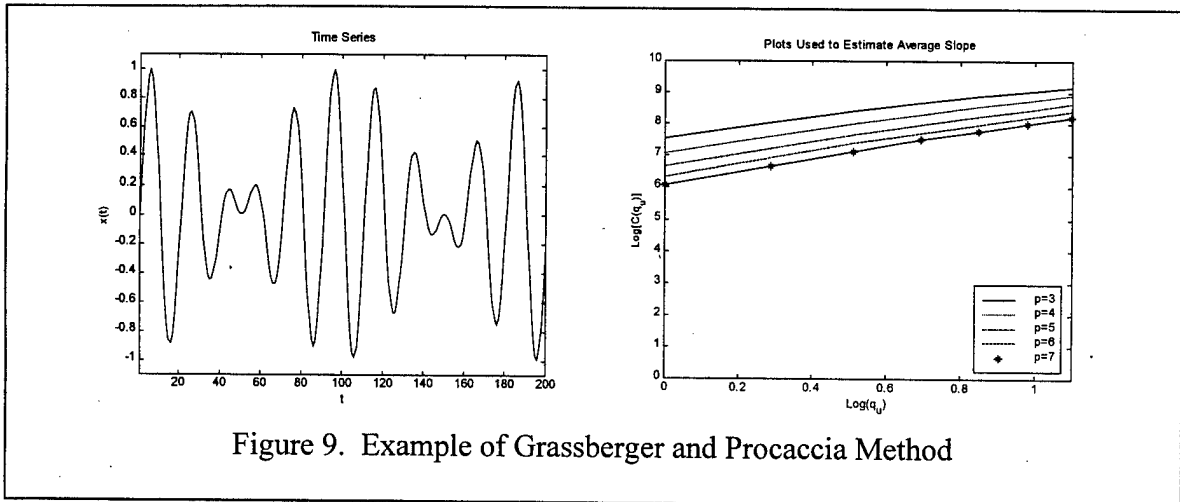
method. For the time series in Figure 9, the mean slope is 1.76. One disadvantage of the Grassberger and Procaccia method is a minimum of 5,000 data points is recommended to adequately determine the fractal dimension of a time series. However, this is not a serious limitation. The example in Figure 9 used only 200 data points of the time series to attain an estimate of $d_f(\mathbf{x}) \approx 1.76$. 5,220 data points of the time series provided an estimate of $d_f(\mathbf{x}) \approx 1.7$ [146].

Once the fractal dimension of a time series is computed, Lapedes recommends applying Taken's Theorem [150, 151] to provide an upper and lower bound on the number of lagged inputs required for a TDNN [76, 77, 78]. The number of lagged network inputs for a TDNN must satisfy

$$f_d(A) < L+1 < 2 \cdot f_d(A)+1 \tag{57}$$

Applying Equation 57 to the example given in Figure 9 results in $1 < L < 3$.

2.5.1.2 Drawbacks of Time Delay Neural Networks (TDNN)

There are several drawbacks to using a TDNN. First, it requires that there be some interface with the world, which buffers the input, so that it can be presented all at once [31]. Second, the shift register imposes a rigid limit on the duration of patterns since the input layer must provide for the longest possible pattern [31]. Finally, and most seriously, a TDNN does not easily distinguish relative temporal position from absolute temporal position [31]. As an illustration of relative and absolute temporal position, consider the following two time series:

$$[0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$[0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]$$

These two time series appear to have the same basic pattern that is displaced in time by two time periods. However, if each of these time series are now treated as a vector for input into a TDNN, the geometric interpretation of the two vectors would show the two vectors to be quite dissimilar and spatially distant. A TDNN can be trained to treat these two vectors as having similar patterns but the similarity as learned by the TDNN is the consequence of an external teacher instead of the similarity structure of the pattern. As a results, the desired similarity will not generalize well to novel patterns.

2.5.2   Recurrent Neural Networks (RNN)

A RNN is another special type of temporal ANN that does not suffer from the shortcomings of an TDNN. A RNN contains an ensemble of interconnected nodes and is capable of learning, recognizing, and classifying temporal data [31]. A RNN allows for the encoding of time by modeling temporal behavior via a feedback or recurrent loop. The so-called recurrent connection weights are set to 1.0 and are not subject to adjustment. The recurrent connections feed back onto the input layer. A RNN uses this feedback to encode and learn these temporal sequences. As such, a RNN can reflect both differences and changes over both time and space. A RNN allows time to be represented by the effect it has on processing. The recurrent connections allow the hidden nodes to see previous hidden and/or output node activations so that subsequent behavior can be shaped by previous responses [31]. These recurrent connections are what give a RNN *memory* [31]. Once a hidden or output node activation is fed back onto the input layer, the recurrent node is termed a *context node*. The context nodes are initially set at 0.0 if

hyperbolic tangent nonlinear transfer functions are used on the hidden nodes and to 0.5 if

sigmoid nonlinear transfer functions are used on the hidden nodes. The three most

popular types of RNNs are the Elman RNN, the Jordan RNN, and the Williams and

Zipser RNN.

2.5.2.1 Elman Recurrent Neural Network (RNN)

The Elman RNN as shown in Figure 10 is the most commonly used RNN. In an

Elman RNN, the hidden layer is fed back onto the input layer with one time delay. Each

connection in an Elman RNN has a weight. An Elman RNN typically has three layers:

- One input layer containing $i = 1,...,I$ input nodes at time t, denoted as $x_i(t)$, $j_0 = 1,...,J$ context nodes (representing the outputs of the $j = 1,...,J$ hidden nodes at time $t-1$) at time $t-1$ denoted as $y_{j_0}(t-1)$, and a bias node x_0.
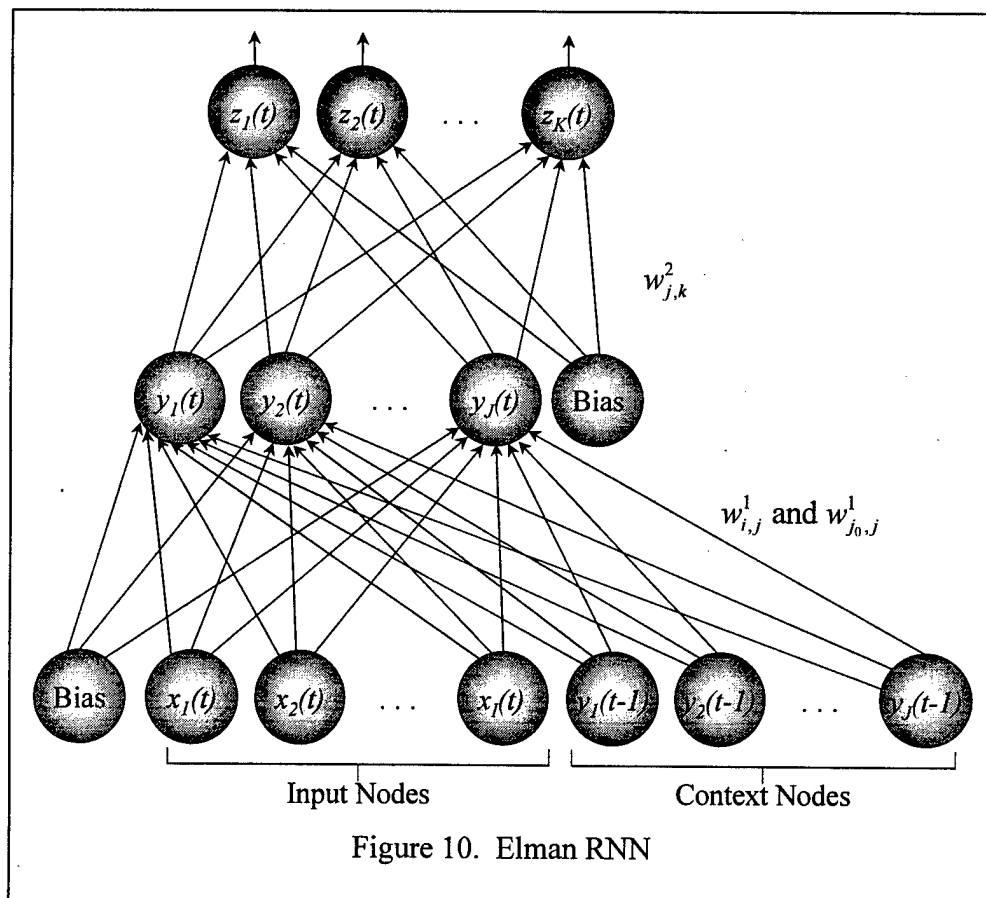
Figure 10. Elman RNN

- One hidden layer containing $j = 1, ..., J$ hidden nodes at time t denoted as $y_j(t)$ and a bias node y_0.
- One output layer containing $k = 1, ..., K$ output nodes at time t denoted as $z_k(t)$.

The Elman RNN as depicted in Figure 10 has a $I + J / J / K$ architecture meaning it has I input nodes and J context nodes on the input layer, J hidden nodes on the hidden layer, and K output nodes on the output layer. The Elman RNN as depicted in Figure 10 is for classification with K output classes. An Elman RNN can be used for either function estimation or function prediction in addition to classification or classification prediction. An Elman RNN has been used to recognize the structure in letter sequences, words, and even simple sentences [31].

The user defines the number of inputs, the number of output classes, and the number of hidden nodes. The activations on the hidden and output nodes can be any of the transfer function as given in Table 1 though in practice, the hidden nodes are typically activated by the hyperbolic tangent nonlinear transfer function and the output nodes are typically activated by the linear transfer function with slope $= 1$ [31]. The output of an Elman RNN is based upon the current inputs and previous internal state which is represented by the context nodes on the input layer. The context nodes are the activations from the hidden nodes at time $t - 1$. After the architecture of an Elman RNN is selected, it is trained via *backpropagation through time* (BPTT) as developed by Rumelhart [128]. BPTT is simply the backpropagation training algorithm in batch. In other words, the actual outputs for all time steps are compared in batch to the desired outputs and the backpropagation of error is used to adjust all of the weights. A disadvantage of BPTT is the memory required due to the temporal component. Whereas a feedforward MLP ANN

58

will have first layer weights $w^1_{i,j}$ for $i = 1,2,...,I$ input features and $j = 1,2,...,J$ hidden nodes, an Elman RNN will have first layer weights $w^1_{i,j}$ for $i = 1,2,...,I$ input features, and $j = 1,2,...,J$ hidden nodes in addition to first layer weights $w^1_{j_0,j}$, for $j_0 = 1,2,...,J$ context nodes and $j = 1,2,...,J$ hidden nodes.

Through training an ERNN, the hidden units develop an internal representation that "recode" input features and the previous internal state [31]. The effect of time is implicit in this internal representations of the hidden nodes [31].

2.5.2.2 Jordan Recurrent Neural Network (RNN)

The Jordan RNN as shown in Figure 11 is another commonly used RNN. In a Jordan RNN, the output layer is fed back onto the input layer with one time delay. Each connection in a Jordan RNN has a weight. A Jordan RNN typically has three layers:

- One input layer containing $i = 1,...,I$ input nodes at time t, denoted as $x_i(t)$, $k_0 = 1,...,K$ context nodes (representing the outputs of the $k = 1,...,K$ output nodes at time $t-1$) at time $t-1$ denoted as $z_{k_0}(t-1)$, and a bias node x_0.
- One hidden layer containing $j = 1,...,J$ hidden nodes at time t denoted as $y_j(t)$ and a bias node y_0.
- One output layer containing $k = 1,...,K$ output nodes at time t denoted as $z_k(t)$.

The Jordan RNN as depicted in Figure 10 has a $I + K / J / K$ architecture meaning it has I input nodes and K context nodes on the input layer, J hidden nodes on the hidden layer, and K output nodes on the output layer. The Jordan RNN as depicted in Figure 11 is for classification with K output classes. A Jordan RNN can be used for either function estimation or function prediction in addition to classification or classification prediction.

Figure 11. Jordan RNN

The user defines the number of inputs, the number of output classes, and the number of hidden nodes. The activations on the hidden and output nodes can be any of the transfer function as given in Table 1 though in practice, the hidden nodes are typically activated by the hyperbolic tangent nonlinear transfer function and the output nodes are typically activated by the linear transfer function with slope $= 1$ [69]. The output of a Jordan RNN is based upon the current inputs and previous outputs which are represented by the context nodes on the input layer. The context nodes are the outputs from the output nodes at time $t-1$. After the architecture of a Jordan RNN is selected, it is trained via BPTT. Whereas a feedforward MLP ANN will have first layer weights $w_{i,j}^1$ for $i = 1,2,...,I$ input features and $j = 1,2,...,J$ hidden nodes, a Jordan RNN will have

60

first layer weights $w_{i,j}^1$ for $i = 1,2,...,I$ input features, and $j = 1,2,...,J$ hidden nodes in addition to first layer weights $w_{k_0,j}^1$, for $k_0 = 1,2,...,K$ context nodes and $j = 1,2,...,J$ hidden nodes.

Through training a Jordan RNN, the hidden units develop an internal representation that "recode" input features and the previous outputs [31]. The effect of time is implicit in this internal representations of the hidden nodes [31].

2.5.2.3 William and Zipser Recurrent Neural Network (RNN)

The Williams and Zipser RNN as shown in Figure 12 is another commonly used RNN. The Williams and Zipser RNN is a combination of the Elman RNN and the Jordan RNN in that both the hidden layer and the output layer are fed back onto the input layer

Figure 12. Williams and Zipser RNN
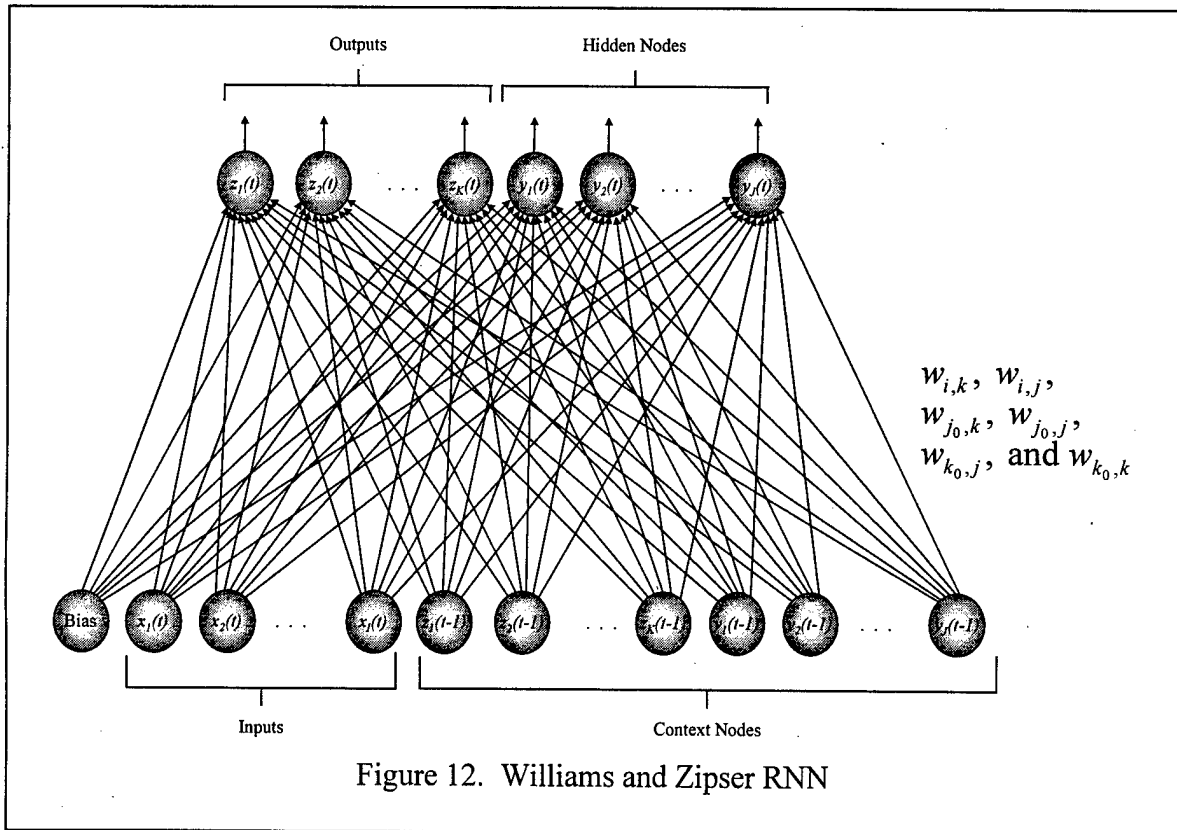
with one time delay. Each connection in a Williams and Zipser RNN has a weight. A Williams and Zipser RNN typically has two layers:

- One input layer containing $i = 1,...,I$ input nodes at time t, denoted as $x_i(t)$, $j_0 = 1,...,J$ context nodes (representing the outputs of the $j = 1,...,J$ hidden nodes at time $t-1$) at time $t-1$ denoted as $y_{j_0}(t-1)$, $k_0 = 1,...,K$ context nodes (representing the outputs of the $k = 1,...,K$ output nodes at time $t-1$) at time $t-1$ denoted as $z_{k_0}(t-1)$, and a bias node x_0.
- One output layer containing $k = 1,...,K$ output nodes at time t denoted as $z_k(t)$ and $j = 1,...,J$ hidden nodes at time t denoted as $y_j(t)$.

The Williams and Zipser RNN as depicted in Figure 12 has a $I + J + K / J + K$ architecture meaning it has I input nodes, J context nodes, and K context nodes on the input layer. It also has J hidden nodes and K output nodes on the output layer. The Williams and Zipser RNN as depicted in Figure 12 is for classification with K output classes. The Williams and Zipser RNN can be used for either function estimation or function prediction in addition to classification or classification prediction. A Williams and Zipser RNN has been used in the past to mimic human behavior and performance [36].

The user defines the number of inputs, the number of output classes, and the number of hidden nodes. The activations on the hidden and output nodes can be any of the transfer function as given in Table 1 though in practice, the hidden nodes are typically activated by the hyperbolic tangent nonlinear transfer function and the output nodes are typically activated by the linear transfer function with slope = 1 [166, 167]. The output of a Williams and Zipser RNN is based upon the current inputs, previous outputs which are represented by the context nodes on the input layer and previous internal state which is also represented by the context nodes on the input layer.. The context nodes are the outputs from the output nodes at time $t-1$ and the activations of the hidden nodes at

time $t-1$. After the architecture of the Williams and Zipser RNN is selected, it is trained. There are several ways to train a Williams and Zipser RN but the most common is the real-time recurrent learning (RTRL) algorithm as developed by Williams and Zipser [166, 167]. Whereas a feedforward MLP ANN will have first layer weights $w_{i,j}^1$ for $i = 1,2,\ldots,I$ input features and $j = 1,2,\ldots,J$ hidden nodes, a Williams and Zipser RNN will have weights $w_{i,j}$ for $i = 1,2,\ldots,I$ input features and $j = 1,2,\ldots,J$ hidden nodes in addition to:

- Weights $w_{i,k}$ for $i = 1,2,\ldots,I$ input features and $k = 1,2,\ldots,K$ output nodes
- Weights $w_{j_0,j}$ for $j_0 = 1,2,\ldots,J$ context nodes and $j = 1,2,\ldots,J$ hidden nodes
- Weights $w_{j_0,k}$ for $j_0 = 1,2,\ldots,J$ context nodes and $k = 1,2,\ldots,K$ output nodes
- Weights $w_{k_0,j}$ for $k_0 = 1,2,\ldots,K$ context nodes and $j = 1,2,\ldots,J$ hidden nodes
- Weights $w_{k_0,k}$ for $k_0 = 1,2,\ldots,K$ context nodes and $k = 1,2,\ldots,K$ output nodes.

63

3.1   Introduction

This chapter provides a literature review of feature saliency measures. Topics covered include the importance of features saliency, rules of thumb for the appropriate number of features, feature saliency measures, and feature screening methods. The feature saliency measures discussed include principal component analysis (PCA), Ruck's partial derivative-based saliency measure, and Tarr's weight-based saliency measure. The feature screening methods discussed are those developed by Setiono-Liu, Belue-Bauer, and Steppe-Bauer. All feature screening methods discussed are *backwards* screening methods in that the heuristics begin with all candidate features and then *remove* features. A *forward* screening method, on the other hand, *adds* features.

3.2   Importance of Feature Saliency

It is well known that the use of too many input features in a classifier can have negative effects. First of all, insignificant input features to a neural network may reduce classification accuracy. Also, too many input features may *overfit* the data resulting in a decreased capability to generalize [110]. Ruck and Rogers improved the CA_{test} for a breast cancer detection problem from 73% to 78% (no standard deviations reported) by removing 14 nonsalient features from 21 candidate input features [125]. Setiono and Liu improved the CA_{test} for the *Monks 3* problem [153] from 93.55% (standard deviation of 1.41) to 98.41% (standard deviation of 1.66) by removing 13 nonsalient features from 17 candidate input features [132]. Shaudys and Leen improved the CA_{test} for a spoken letter

recognition system by removing 146 nonsalient features from 160 candidate input features [133].

In addition, the so-called "Curse of Dimensionality" states that as the number of features grows, the number of training vectors required grows exponentially [25]. Thus, the number of training vectors required can be reduced which, in turn, typically reduces the training time if nonsalient features are removed.

3.3 Rules of Thumb

Both Foley's Rule and Cover's Theorem provide rules of thumb for the appropriate number of input features.

3.3.1 Foley's Rule

Foley's Rule states that if the number of training exemplars M_{train} is greater than 3 times the number of features I times the number of classes K, then the training set error is approximately the test set error [37]. In addition, the test set error is close to the optimum error attained by a Bayes classifier [37]. In equation form, Foley's Rule for normally distributed inputs is:

$$M_{train} > 3 \cdot I \cdot K \tag{58}$$

Since Foley's Rule assumes that the features are normally distributed, a greater ratio should be used if the distribution of the features is unknown [37]. In equation form, Foley's Rule for inputs with unknown distributions becomes:

$$M_{train} \gg 3 \cdot I \cdot K \tag{59}$$

65

3.3.2 Cover's Theorem

Cover states that the training set error will be near zero if the total number of training exemplars M_{train} for a two-class problem is more than twice the number of features I [23]. In equation form, Cover's Theorem is:

$$M_{train} > 2 \cdot I \tag{60}$$

Cover's Theorem is valid regardless of the distribution of the features [23]. Note that Foley's Rule is more stringent than Cover's Theorem. If Foley's Rule is satisfied for a two-class problem, then Cover's Theorem is also satisfied.

Cover's Theorem also has extensions for providing an upper bound on the total number of hidden nodes J [23] so that:

$$J < \frac{0.5 \cdot I - 1}{M_{train} + 1} \tag{61}$$

3.4 Feature Saliency Measures

PCA is a classic statistical method of dimensionality reduction that can be applied to ANNs but with some drawbacks. The majority of feature saliency measures for use in ANNs have been developed in the last decade. Of these feature saliency measures, two general categories exist: partial derivative-based saliency measures and weight-based saliency measures.

3.4.1 Principal Component Analysis (PCA)

PCA is a classical statistical method used on multivariate data sets in order to reduce the dimensionality. The use of PCA, also referred to as the Karhunen-Loève transformation [71], for feature saliency does not require a trained ANN. In addition,

PCA does not require any information on the classes to be discriminated. The only thing needed to calculate the principal components of the input feature set is the covariance matrix of the input data [26]. The underlying assumption of PCA is that the covariance of the input feature set is the most important information characteristic of the data [149]. While this assumption is suitable for reconstruction of the input feature set, it is not appropriate for classification. The principal components (PC), which are linear combinations of the input features, capture as much of the data variability as possible in a linear fashion. PCA may be suitable for feature saliency if the ANN is used for function estimation or function prediction [149]. But PCA is not appropriate if the ANN is used for classification where the objective is to maintain separation of the classes and not necessarily to explain the variance [149]. Following the procedures outlined in Dillon and Goldstein [26], the first step in PCA is to *mean correct* the normalized input feature set following:

$$\widetilde{\mathbf{X}}' = \begin{bmatrix} x'_{1,1} - \overline{x}'_1 & x'_{1,2} - \overline{x}'_1 & \cdots & x'_{1,M_{train,test}} - \overline{x}'_1 \\ x'_{2,1} - \overline{x}'_2 & x'_{2,2} - \overline{x}'_2 & \cdots & x'_{2,M_{train,test}} - \overline{x}'_2 \\ \vdots & \vdots & \ddots & \vdots \\ x'_{I,1} - \overline{x}'_I & x'_{I,2} - \overline{x}'_I & \cdots & x'_{I,M_{train,test}} - \overline{x}'_I \end{bmatrix} \tag{62}$$

where $\widetilde{\mathbf{X}}'$ is the mean corrected normalized input feature set and $\overline{x}'_i$ is the mean of feature x'_i for $i = 1,2,\ldots,I$. $\overline{x}'_i$ is calculated following Equation 17. The exemplars from the training and test sets are included if the input feature set was normalized using Equation 19 or 20. If the training and test sets were normalized separately, replace $M_{train,test}$ in Equation 62 with M_{train} to mean correct the training set and with M_{test} to mean correct the test set. The covariance matrix $\mathbf{C}$ is calculated using $\widetilde{\mathbf{X}}'$ as:

67

$$C = \frac{\left(\tilde{\mathbf{X}}'\right)^{\mathrm{T}} \cdot \tilde{\mathbf{X}}' - \dfrac{1}{M_{train,test}} \cdot \left(\tilde{\mathbf{X}}' \cdot \mathbf{1}\right)^{\mathrm{T}} \cdot \left(\mathbf{1}^{\mathrm{T}} \cdot \tilde{\mathbf{X}}'\right)}{M_{train,test} - 1} \tag{63}$$

If the training and test sets were normalized separately, replace $M_{train,test}$ in Equation 63 with M_{train} to mean correct the training set and with M_{test} to mean correct the test set. The eigenvalues and normalized eigenvectors are next extracted from $\mathbf{C}$ so that

$$\mathbf{C} = \mathbf{P} \cdot \Lambda \cdot \mathbf{P}' \tag{64}$$

where $\mathbf{P}$ is an $I \times I$ orthogonal matrix whose columns are the normalized eigenvectors of $\mathbf{C}$ and Λ is an $I \times I$ diagonal matrix whose diagonal elements $\lambda_{i,i}$ for $i = 1,2,\ldots,I$ are the eigenvalues of $\mathbf{C}$. Now the PCs can be computed as:

$$\mathbf{PC} = \begin{bmatrix} \mathbf{PC}_1 & \mathbf{PC}_2 & \cdots & \mathbf{PC}_I \end{bmatrix} = \mathbf{P}' \cdot \tilde{\mathbf{X}} \tag{65}$$

The PCs are uncorrelated. The first PC denoted as $\mathbf{PC}_1$ accounts for the largest amount of variance and is the ith column in $\mathbf{PC}$ that corresponds to the largest eigenvalue $\lambda_{i,i}$ in Λ. The second PC denoted as $\mathbf{PC}_2$ accounts for the second largest amount of variance and is the ith column in $\mathbf{PC}$ that corresponds to the second largest eigenvalue $\lambda_{i,i}$ in Λ. And so on.

After the PCs are computed, the PCs are then used as inputs to an ANN. The PCs can be rank ordered using the eigenvalues where higher eigenvalues correspond to higher saliency and lower eigenvalues correspond to lower saliency.

3.4.2 Partial Derivative-Based

Ruck developed a partial derivative-based saliency measure that is based upon the sensitivity of an ANN's outputs to an input and utilizes the partial derivative of the

outputs with respect to a specific input [124, 126]. The partial derivative-based saliency measure depends on the distribution of the input data relative to the decision boundaries. The partial derivative-based saliency measure can be used to rank order the features from most salient to least salient where higher partial derivative-based saliency measure values indicate higher relative saliency and lower values indicate lower relative saliency. The partial derivative-based saliency measure determines feature i's effect on the ANN's outputs by calculating the sum of absolute value of the partial derivatives of the outputs with respect to feature i. This partial derivative-based saliency measure depends upon the inputs and the weights within the trained ANN. Ruck derived two types of partial derivative-based saliency measures:

- One that calculates the partial derivatives using the normalized (or standardized) training exemplars.
- One that calculates the partial derivatives using pseudo-sampling of the input feature space [124, 126].

3.4.2.1 Partial Derivative-Based Saliency Measure

The most commonly used partial derivative-based saliency measure is the one that calculates the partial derivatives at the normalized (or standardized) training exemplars as:

$$\Lambda_i = \frac{1}{K} \cdot \frac{1}{M_{train}} \sum_{k=1}^{K} \sum_{m=1}^{M_{train}} \left| \frac{\partial z_{k,m}(\mathbf{x}'_m, \mathbf{W})}{\partial x'_{i,m}} \right| \tag{66}$$

where Λ_i is the partial derivative-based saliency measure for feature $i = 1,2,...,I$ and $z_{k,m}(\mathbf{x}'_m, \mathbf{W})$ is the actual output of output node k with input exemplar $\mathbf{x}'_m$ for $m = 1,2,...,M_{train}$ with the trained ANN weight matrix $\mathbf{W}$. $z_{k,m}(\mathbf{x}'_m, \mathbf{W})$ is used in this

derivation of the partial derivative-based saliency measure instead of $z_{k,m}$ to clearly annotate that the calculated partial derivatives are functions of $\mathbf{x}'_m$ for $m = 1,2,\ldots,M_{train}$ and $\mathbf{W}$. The normalized (or standardized) training exemplars $x'_{i,m}$ for $i = 1,2,\ldots,I$ and for $m = 1,2,\ldots,M_{train}$ are used so that the features are "unitless" thus preventing the input features with larger value from dominating. The partial derivative-based saliency measure as given in Equation 66 is appropriate if the training set adequately represents the input feature space and in particular, the boundaries separating the classes. Using Equation 29, the partial derivatives are calculated as:

$$\frac{\partial z_{k,m}(\mathbf{x}'_m, \mathbf{W})}{\partial x'_{i,m}} = \frac{\partial}{\partial x'_{i,m}}\left[f_k\left(w^2_{0,k} + \sum_{j=1}^{J} w^2_{j,k} \cdot y_{j,m}(\mathbf{x}'_m, \mathbf{W}) \right) \right] \tag{67}$$

$$\frac{\partial z_{k,m}(\mathbf{x}'_m, \mathbf{W})}{\partial x'_{i,m}} = \dot{f}_k\left(w^2_{0,k} + \sum_{j=1}^{J} w^2_{j,k} \cdot y_{j,m}(\mathbf{x}'_m, \mathbf{W}) \right) \cdot \frac{\partial}{\partial x'_{i,m}}\left(w^2_{0,k} + \sum_{j=1}^{J} w^2_{j,k} \cdot y_{j,m}(\mathbf{x}'_m, \mathbf{W}) \right) \tag{68}$$

Substituting Equation 32 into Equation 68:

$$\frac{\partial z_{k,m}(\mathbf{x}'_m, \mathbf{W})}{\partial x'_{i,m}} = \dot{z}_{k,m}(\mathbf{x}'_m, \mathbf{W}) \cdot \frac{\partial}{\partial x'_{i,m}}\left(\sum_{j=1}^{J} w^2_{j,k} \cdot y_{j,m}(\mathbf{x}'_m, \mathbf{W}) \right) \tag{69}$$

Knowing that

$$y_{j,m}(\mathbf{x}'_m, \mathbf{W}) = f_j\left(w^1_{0,j} + \sum_{i=1}^{I} w^1_{i,j} \cdot x'_{i,m} \right) \tag{70}$$

Equation 69 becomes:

$$\frac{\partial z_{k,m}(\mathbf{x}'_m, \mathbf{W})}{\partial x'_{i,m}} = \dot{z}_{k,m}(\mathbf{x}'_m, \mathbf{W}) \cdot \frac{\partial}{\partial x'_{i,m}}\left[\sum_{j=1}^{J} w^2_{j,k} \cdot f_j\left(w^1_{0,j} + \sum_{i=1}^{I} w^1_{i,j} \cdot x'_{i,m} \right) \right]$$

$$= \dot{z}_{k,m}(\mathbf{x}'_m, \mathbf{W}) \cdot \sum_{j=1}^{J} w^2_{j,k} \cdot \frac{\partial}{\partial x'_{i,m}}\left[f_j\left(w^1_{0,j} + \sum_{i=1}^{I} w^1_{i,j} \cdot x'_{i,m} \right) \right]$$

$$\frac{\partial z_{k,m}(\mathbf{x}'_m,\mathbf{W})}{\partial x'_{i,m}} = \dot{z}_{k,m}(\mathbf{x}'_m,\mathbf{W}) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{f}_j \left(w_{0,j}^1 + \sum_{i=1}^{I} w_{i,j}^1 \cdot x'_{i,m} \right) \cdot \frac{\partial}{\partial x'_{i,m}} \left(w_{0,j}^1 + \sum_{i=1}^{I} w_{i,j}^1 \cdot x'_{i,m} \right) \quad (71)$$

For clarity, let

$$\dot{y}_j(\mathbf{x}'_m,\mathbf{W}) = \dot{f}_j \left(w_{0,j}^1 + \sum_{i=1}^{I} w_{i,j}^1 \cdot x'_{i,m} \right) \qquad (72)$$

Substituting Equation 72 into Equation (71:

$$\frac{\partial z_{k,m}(\mathbf{x}'_m,\mathbf{W})}{\partial x'_{i,m}} = \dot{z}_{k,m}(\mathbf{x}'_m,\mathbf{W}) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{y}_{j,m}(\mathbf{x}'_m,\mathbf{W}) \cdot \frac{\partial}{\partial x'_{i,m}} \left(w_{0,j}^1 + \sum_{i=1}^{I} w_{i,j}^1 \cdot x'_{i,m} \right)$$

$$\frac{\partial z_{k,m}(\mathbf{x}'_m,\mathbf{W})}{\partial x'_{i,m}} = \dot{z}_{k,m}(\mathbf{x}'_m,\mathbf{W}) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{y}_{j,m}(\mathbf{x}'_m,\mathbf{W}) \cdot w_{i,j}^1 \qquad (73)$$

Substituting Equation 73 into Equation 66,

$$\Lambda_i = \frac{1}{K} \cdot \frac{1}{M_{train}} \sum_{k=1}^{K} \sum_{m=1}^{M_{train}} \left| \dot{z}_{k,m}(\mathbf{x}'_m,W) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{y}_{j,m}(\mathbf{x}'_m,W) \cdot w_{i,j}^1 \right| \qquad (74)$$

Those exemplars that are closest to the boundaries separating the classes will contribute the most to the calculation of Equation 74.

3.4.2.2 Partial Derivative-Based Saliency Measure with Pseudo-Sampling

The other method for calculating the partial derivative-based saliency measure uses a pseudo-sampling technique on the I-dimensional feature space. The pseudo-sampling technique uniformly divides the I-dimensional feature space into $r = 1,2,\ldots,R^I$ range bins. Instead of calculating the partial derivatives at the normalized (or standardized) training exemplars $\mathbf{x}'_m$ for $m = 1,2,\ldots,M_{train}$, the partial derivatives are calculated at the midpoints of each range bin denoted as $\mathbf{x}'_r$ for $r = 1,2,\ldots,R^I$. The partial derivative-based saliency measure with pseudo-sampling can be written as a

function of the range bin midpoints $\mathbf{x}'_r$ for $r = 1,2,\dots,R^I$ and $\mathbf{W}$ as follows:

$$\hat{\Lambda}_i = \frac{1}{K} \cdot \frac{1}{R^I} \sum_{k=1}^{K} \sum_{r=1}^{R^I} \left| \frac{\partial z_{k,r}(\mathbf{x}'_r, \mathbf{W})}{\partial x'_{i,r}} \right| \qquad (75)$$

where $\hat{\Lambda}_i$ is the partial derivative-based saliency measure with pseudo-sampling for feature $i = 1,2,\dots,I$ and $x'_{i,r}$ is the normalized range bin midpoints for input $i = 1,2,\dots,I$ and range bin $r = 1,2,\dots,R^I$. Note that the normalized (or standardized) range bin midpoints $x'_{i,r}$ for $i = 1,2,\dots,I$ and for $r = 1,2,\dots,R^I$ are used so that the features are "unitless" thus preventing the input features with larger value from dominating. Following the logic of the derivations for the partial derivative-based saliency measure without pseudo-sampling, Equation 75 can be rewritten as:

$$\hat{\Lambda}_i = \frac{1}{K} \cdot \frac{1}{R^I} \sum_{k=1}^{K} \sum_{r=1}^{R^I} \left| \dot{z}_{k,r}(\mathbf{x}'_r, \mathbf{W}) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{y}_{j,r}(\mathbf{x}'_r, \mathbf{W}) \cdot w_{i,j}^1 \right| \qquad (76)$$

Those exemplars that are closest to the boundaries separating the classes will contribute the most to the calculation of Equation 76. The idea of the pseudo-sampling method is to provide a means to adequately represent the input feature space and, in particular, the boundaries between classes. However, empirical evidence provided by Steppe shows that the rankings provided by the partial derivative-based saliency measure appear to be similar to that provided by the partial derivative-based saliency measure with pseudo-sampling [136]. This conclusion makes sense since the partial derivatives, regardless if made at the training exemplars or at the pseudo-sampled points, are dependent upon the weights that were trained using the training exemplars. The weights of a trained ANN represent the boundaries as approximated from the training exemplars.

72

3.4.3 Weight-Based Saliency Measure

Tarr developed a weight-based saliency measure that determines the saliency of feature i by summing the squared values of the first layer weights connecting feature i to the hidden nodes [152]. Tarr describes the idea behind the weight-based saliency measure:

> When a weight is updated, the network moves the weight a small amount based on the error. Given that a particular feature is relevant to the problem solution, the weight would be moved in a constant direction until a solution with no error is reached. If the error term is consistent, the direction of the movement of the weight vector, which forms a hyperplane decision boundary, will also be consistent . . . In a similar fashion, if the feature did not contribute to a solution, the weight updates would be random. In other words, useful features would cause the weights to grow, while weights attached to nonsalient features simply fluctuate around zero. [152: 44-45]

The weight-based saliency measure can be written as follows:

$$\tau_i = \sum_{j=1}^{J} (w_{i,j}^1)^2 \tag{77}$$

where τ_i is the weight-based saliency measure for feature $i = 1,2,....I$ [152]. Equation 77 is simply the sum of the squared weights between input node i and hidden node j. There are three variants to the weight-based saliency measure. The first variant is the Euclidean norm of the weights of the feature and is typically equated by the square root of the sum of the squared weights emanating from a given input node and can be written as:

$$\tau_i^{v1} = \sqrt{\sum_{j=1}^{J} (w_{i,j}^1)^2} \tag{78}$$

where τ_i^{v1} is the first variant of τ_i [111, 136]. The second variant as used by Reinhart is the "taxi-cab" norm of the weights of the features and is calculated by summing the

absolute value of the weights emanating from a given input node as:

$$\tau_i^{v2} = \sum_{j=1}^{J} \left| w_{i,j}^1 \right| \tag{79}$$

where τ_i^{v2} is the second variant of τ_i [111, 136]. The third variant is the infinity norm of the weights of the feature and is simply the largest weight in absolute value of all the weights from a given input node:

$$\tau_i^{v3} = \max \left\{ \left| w_{i,j}^1 \right| : j = 1,2,\ldots,J \right\} \tag{80}$$

where τ_i^{v3} is the third variant of τ_i [111].

Utilizing the triangle inequality [2], Steppe derived the theoretical relationship between the partial derivative-based saliency measure in Equation 74 and the second variant of the weight-based saliency measure in Equation 79 as:

$$\Lambda_i \leq \Phi' \cdot \tau_i^{v2} \tag{81}$$

where Φ is a vector of constants [136].

Note that the input features must be normalized or standardized when using any of the weight-based saliency measures.

3.5 Feature Screening

Feature screening methods provide a way to determine the parsimonious set of salient features while maintaining good classification accuracy. Whereas feature saliency measures help to rank order input features, feature screening methods provide a means to remove irrelevant and/or redundant input features. The feature screening method developed by Setiono and Liu utilizes a penalty term on the error function while training a feedforward MLP ANN in order to distinguish irrelevant input features using a weight-

74

based saliency measure. The Setiono-Liu screening method uses *CA* as a MOE. The feature screening method developed by Belue and Bauer utilizes an injected noise feature. Salient features are distinguished from nonsalient features using confidence intervals around the saliency measure (both a partial derivative-based saliency measure and a weight base saliency measure are presented). Confidence intervals for the saliency measures are attained from training the feedforward MLP ANN at least 30 times.. The feature screening method developed by Steppe and Bauer improves upon that of Belue and Bauer by incorporating either a paired *t*-test or a Bonferroni joint hypothesis test in addition to reducing the number of trained ANNs required to at least 10.

3.5.1 *Error Term Penalty Function*

The screening method proposed by Setiono and Liu provides a method for removing irrelevant and redundant features in a feedforward MLP ANN [132]. The Setiono-Liu screening method assumes that the important information for determining feature saliency lies in the first layer weights [132]. Their screening method augments the error term by an adaptive penalty function on the first layer weights of the form:

$$Penalty_{i,j} = \frac{\epsilon_1 \cdot \beta \cdot \left(w_{i,j}^1\right)^2}{1 + \beta \cdot w_{i,j}^1} + \epsilon_2 \cdot \left(w_{i,j}^1\right)^2 \qquad (82)$$

where ϵ_1, ϵ_2, and β are user-defined parameters [132]. A plot of the recommended initial penalty function with $\epsilon_1 = 0.1$, $\epsilon_2 = 0.0001$, and $\beta = 10$ is shown in Figure 13. The penalty function serves as a complexity measure of a feedforward MLP ANN that was developed for use as a network pruning algorithm [131]. The objective of Equation 82 is to force as many first layer weights as possible to zero during training since a

Figure 13. Setiono-Liu Penalty Function on the First Layer Weights
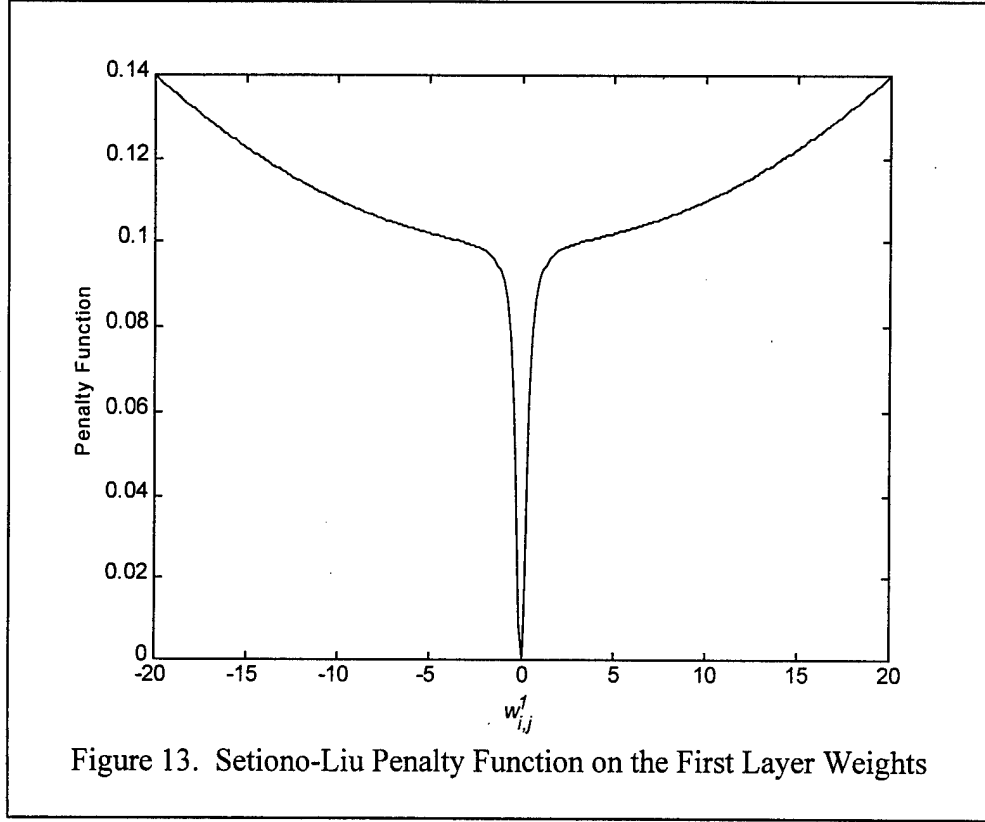
weight with a small magnitude will incur almost no penalty. On the other hand, a weight with a large magnitude will incur a penalty during training that increases as a quadratic function of the weight's magnitude. The user-defined parameters $\in_1$, $\in_2$, and β determine the range over which the value of the penalty function is approximately equal to $\in_1$.

Setiono-Liu Screening Method

1. Set the allowable maximum decrease in CA_{test} denoted as ΔCA_{test} ($\Delta CA_{test} = 3\%$ is recommended).

2. Normalize or standardize the input features. Separate the input feature set into a training set and a test set.

3. Initialize the weights. Train a feedforward MLP ANN with all available features. Compute CA_{train} and CA_{test} following Equations 11 and 12 .

76

4. For $i = 1, 2, \ldots, I$, drop feature x_i from the ANN by setting $w_{i,j}^1 = 0 \; \forall \; j = 1, 2, \ldots J$. Compute CA_{test} without feature x_i, denoted as CA_{test}^i, following Equations 11 and 12.

5. Determine $\max_i \left(CA_{test}^i \right)$ and the average of CA_{test}^i denoted as $\overline{CA}_{test}^i$ where

$$\overline{CA}_{test}^i = \frac{1}{I} \cdot \sum_{i=1}^{I} CA_{test}^i \tag{83}$$

6. If $CA_{test} - \max \left(CA_{test}^i \right) \le \Delta CA_{test}$:
   a. Drop the feature and the first layer weights associated with the highest CA_{test}^i.
   b. Set $I = I - 1$.
   c. Set $CA_{test} = \max \left\{ CA_{test}, \max \left(CA_{test}^i \right) \right\}$
   d. Update penalty parameters for all features $i = 1, 2, \ldots, I$ with $CA_{test}^i \ge \overline{CA}_{test}^i$ so that $\in_1 (i) = 1.1 \cdot \in_1 (i)$ and $\in_2 (i) = 1.1 \cdot \in_2 (i)$.
   e. Update penalty parameters for all features $i = 1, 2, \ldots, I$ with $CA_{test}^i < \overline{CA}_{test}^i$ so that $\in_1 (i) = \dfrac{\in_1 (i)}{1.1}$ and $\in_2 (i) = \dfrac{\in_2 (i)}{1.1}$.
   f. Go to Step 3.

7. Else stop.

The Setiono-Liu screening method produced good results in many classification problems using the following classification data sets: Monks, IBM, Wisconsin Breast Cancer, United States Congressional Voting Records, Pima Indians Diabetes, and Sonar Targets [132].

3.5.2 Injecting Noise

The Belue-Bauer screening method was the first to inject a noise-like feature into a feedforward MLP ANN to use as a baseline for determining feature saliency [11, 12]. This approach can use either the partial derivative-based saliency measure with pseudo-sampling $\hat{\Lambda}_i$ in Equation 76 or the weight-based saliency measure τ_i in Equation 77.

The Belue-Bauer screening method trains at least 30 ANNs so that an upper one-sided confidence interval may be constructed around the saliency measure of the injected noise feature [11, 12]. A feature with an average saliency measure that falls within this upper one-sided confidence interval is considered to be noise-like or nonsalient. The distribution of the average saliency of a feature is assumed to be normally distributed based upon the Central Limit Theorem (CLT) [88].

<u>Belue-Bauer Screening Method</u>

1. Determine the total number of training sessions $G \geq 30$. Set $g = 1$.

2. Determine the level of significance α.

3. Augment the feature input set with a uniformly distributed U(0,1) noise feature N.

4. Normalize all input features.

5. Separate the input feature set into a training set and a test set.

6. Initialize the weights.

7. Train a feedforward MLP ANN with all available features.

8. Compute the saliency of feature $i = 1,2,\ldots,I$ for training session $g = 1,2,\ldots,G$ using a partial derivative-based saliency measure with pseudo-sampling denoted as $\hat{\Lambda}_{i,g}$ following Equation 76 or using a weight-based saliency measure $\tau_{i,g}$ following Equation 77.

9. Compute the saliency of the injected noise feature N for training session $g = 1,2,\ldots,G$ denoted as $\hat{\Lambda}_{N,g}$ following Equation 76 or $\tau_{N,g}$ following Equation 77

10. If $g < G$, set $g = g + 1$ and go to Step 5.

11. Calculate the observed average saliency of feature $i = 1,2,\ldots,I$ denoted as $\overline{\Lambda}_i$ or $\overline{\tau}_i$ such that:

78

$$\overline{\hat{\Lambda}}_i = \frac{1}{G} \cdot \sum_{g=1}^{G} \hat{\Lambda}_{i,g} \quad \text{or} \quad \overline{\tau}_i = \frac{1}{G} \cdot \sum_{g=1}^{G} \tau_{i,g} \tag{84}$$

12. Calculate the observed average saliency of the injected noise feature N denoted as $\overline{\hat{\Lambda}}_N$ or $\overline{\tau}_N$ in a fashion similar to Equation 84.

13. Compute the one-sided upper $100\% \cdot (1-\alpha)$ confidence interval (CI) for the expected saliency for the injected noise feature denoted as $\mu_{\hat{\Lambda}_N}$ or μ_{τ_N} such that:

$$\mu_{\hat{\Lambda}_N} < \overline{\hat{\Lambda}}_N + t_{\alpha,G-1} \cdot \left(\frac{S_{\hat{\Lambda}_N}}{\sqrt{G}} \right) \quad \text{or} \quad \mu_{\tau_N} < \overline{\tau}_N + t_{\alpha,G-1} \cdot \left(\frac{S_{\tau_N}}{\sqrt{G}} \right) \tag{85}$$

where $t_{\alpha,G-1}$ is the t-value for level of significance α and $G-1$ degrees of freedom and $S_{\hat{\Lambda}_N}$ is the sample standard deviation of $\hat{\Lambda}_{N,g}$ for $g = 1,2,\ldots,G$ computed as:

$$S_{\hat{\Lambda}_N} = \sqrt{\frac{\sum_{g=1}^{G} \left(\hat{\Lambda}_{N,g} - \overline{\hat{\Lambda}}_N \right)^2}{G-1}} \tag{86}$$

S_{τ_N} is the sample standard deviation of $\tau_{N,g}$ for $g = 1,2,\ldots,G$ and is computed in the same fashion as Equation 86.

14. Select those features $i \in \{1,2,\ldots,I\}$ whose average saliency $\overline{\hat{\Lambda}}_N$ or $\overline{\tau}_N$ falls outside the CI computed in Step 12. In other words, retain feature $i \in \{1,2,\ldots,I\}$ if:

$$\overline{\hat{\Lambda}}_i > \overline{\hat{\Lambda}}_N + t_{\alpha,G-1} \cdot \left(\frac{S_{\hat{\Lambda}_N}}{\sqrt{G}} \right) \quad \text{or} \quad \overline{\tau}_i > \overline{\tau}_N + t_{\alpha,G-1} \cdot \left(\frac{S_{\tau_N}}{\sqrt{G}} \right) \tag{87}$$

15. Train a feedforward MLP ANN using the selected features.

The Belue-Bauer screening method produced good results in a *noisy* XOR classification problem (*noisy* in that there are four added *distractor* features which are

known to be nonsalient) and a four-class problem with bivariate normally distributed input features [11, 12].

3.5.3 Improvements to Injecting Noise

The Steppe-Bauer screening method improves on the statistical test performed by Belue-Bauer screening method by using a paired t-test or a conservative Bonferroni joint test to compare the saliency for feature $i = 1,2,...,I$ to that for an injected noise feature [136, 137, 140]. The Steppe-Bauer screening method trains at least 10 ANNs so that the paired t-test or Bonferroni joint test can be conducted [136, 137, 140]. The approach can use either the partial derivative-based saliency measure Λ_i in Equation 74 or the first variant of the weight-based saliency measure τ_i^{v1} in Equation 78 [136, 137, 140]. Like the paired t-test, the Bonferroni joint test is a paired test since it also is performed by comparing *pairs* of observed averages. A paired test is necessary since the saliency values can be different from problem to problem and thus, the magnitude of the injected noise feature should be characterized for the problem at hand [136, 140]. In addition, feature saliency measures computed within the same feedforward MLP ANN are likely to be correlated [136, 140]. Finally, the saliency for feature $i = 1,2,...,I$ and that for an injected noise feature are dependent [136, 140]. The Bonferroni joint test is more conservative than the paired t-test. The Bonferroni joint test utilizes a *family* level of significance $\frac{\alpha}{I}$ whereas the paired t-test uses level of significance α.

<u>Steppe-Bauer Screening Method</u>

1. Determine the total number of training sessions $G \geq 10$. Set $g = 1$.

2. Determine the level of significance α.

3. Augment the feature input set with a uniformly distributed $U(0,1)$ noise feature N.

4. Normalize all input features.

5. Separate the input feature set into a training set and a test set.

6. Initialize the weights.

7. Train a feedforward MLP ANN with all available features.

8. Compute the saliency for feature $i = 1,2,...,I$ for training session g using a partial derivative-based saliency measure denoted as $\Lambda_{i,g}$ following Equation 74 or using a variant of a weight-based saliency measure $\tau_{i,g}^{v1}$ following Equation 78.

9. Compute the saliency for the injected noise feature N for training session g denoted as $\Lambda_{N,g}$ following Equation 74 or $\tau_{N,g}^{v1}$ following Equation 78.

10. Compute the difference $D_{i,g}$ between the saliency value for feature $i = 1,2,...,I$ and that for the injected noise feature N for training session $g = 1,2,...,G$ so that:

$$D_{i,g} = \Lambda_{i,g} - \Lambda_{N,g} \text{ or } D_{i,g} = \tau_{i,g}^{v1} - \tau_{N,g}^{v1} \tag{88}$$

11. If $g < G$, set $g = g + 1$ and go to Step 5.

12. Calculate the observed average difference $\overline{D}_i$ between the expected saliency value for feature $i = 1,2,...,I$ and that for the injected noise feature N as:

$$\overline{D}_i = \frac{1}{G} \cdot \sum_{g=1}^{G} D_{i,g} \tag{89}$$

13. Calculate the sample standard deviation of $\overline{D}_i$ denoted as $S_{\overline{D}_i}$:

$$S_{D_i} = \sqrt{\frac{\sum_{g=1}^{G}\left(D_{i,g} - \overline{D}_i\right)^2}{G-1}} \tag{90}$$

81

Perform $i = 1, 2, \ldots, I$ paired t-tests and Bonferroni joint tests on the following hypothesis:

$$H_0: \quad \mu_{D_i} = 0$$
$$H_a: \quad \mu_{D_i} > 0$$

where μ_{D_i} is the expected difference between the saliency value for feature $i = 1, 2, \ldots, I$ and that for the injected noise feature N such that:

$$\mu_{D_i} = \mu_{\Lambda_i} - \mu_{\Lambda_N} \quad \text{or} \quad \mu_{D_i} = \mu_{\tau_i^{v1}} - \mu_{\tau_N^{v1}} \tag{91}$$

where μ_{Λ_i} or $\mu_{\tau_i^{v1}}$ denote the expected saliency for feature $i = 1, 2, \ldots, I$ and μ_{Λ_N} or $\mu_{\tau_N^{v1}}$ denote the expected saliency for the inject noise feature N. Compute the t-test statistic for feature $i = 1, 2, \ldots, I$ denoted as $t_{s,i}$ is calculated as:

$$t_{s,i} = \frac{\overline{D}_i}{S_{\overline{D}_i} / \sqrt{G}} \tag{92}$$

For $i = 1, 2, \ldots, I$, reject H_0 if $t_{s,i} > t_{\alpha, G-1}$ for the paired t-test.
For $i = 1, 2, \ldots, I$, reject H_0 if $t_{s,i} > t_{\frac{\alpha}{I}, G-1}$ for the Bonferroni joint test.

14. Using either the paired t-test or the Bonferroni joint test, select those features $i = 1, 2, \ldots, I$ that reject H_0.

15. Train a feedforward MLP ANN using the selected features.

The Steppe-Bauer screening method requires that the saliency $\Lambda_{i,g}$ or $\tau_{i,g}^{v1}$ for feature $i = 1, 2, \ldots, I$ for training session $g = 1, 2, \ldots, G$ and the saliency $\Lambda_{N,g}$ or $\tau_{N,g}^{v1}$ for the injected noise feature N for training session $g = 1, 2, \ldots, G$ are normally distributed random variables [136, 137, 140]. If not, the conditions of the Central Limit Theorem (CLT) should be applied. An assumption of normality via the CLT is reasonable for the

partial derivative-based saliency measure $\Lambda_{i,g}$ or $\Lambda_{N,g}$ so long as $M_{train} \geq 30$. An assumption of normality via the CLT is also reasonable when $G \geq 30$. The Steppe-Bauer screening method requires that the saliency $\Lambda_{i,g}$ or $\tau_{i,g}^{v1}$ for feature $i = 1,2,\dots,I$ for training session $g = 1,2,\dots,G$ is identically and independently distributed (IID) with expected saliency μ_{Λ_i} or $\mu_{\tau_i^{v1}}$ and variance $\sigma_{\Lambda_i}^2$ or $\sigma_{\tau_i^{v1}}^2$. Likewise, the saliency $\Lambda_{N,g}$ or $\tau_{N,g}^{v1}$ for the injected noise feature N for training session $g = 1,2,\dots,G$ is assumed to be IID with constant expected saliency μ_{Λ_N} or $\mu_{\tau_N^{v1}}$ and variance $\sigma_{\Lambda_N}^2$ or $\sigma_{\tau_N^{v1}}^2$.

The Steppe-Bauer screening method produced good results in a noisy version of Fisher's iris classification problem and a real-world armor piercing incendiary projectile classification problem [136, 137, 140].

3.6   Backwards Screening versus Forward Screening

The main advantage of backwards screening methods over forward screening methods is computational efficiency. In a backwards screening, an ANN is first trained with I candidate input features for E_{max} epochs. One feature is removed and the same ANN or perhaps a new ANN is trained with the remaining $I-1$ input features for E_{max} epochs. Another feature is removed and the same ANN or perhaps a new ANN is trained with the remaining $I-2$ input features for E_{max} epochs. And so on until all candidate input features are exhausted. A total of I ANNs are trained in a backwards screening method. A MOE such as CA_{valid} may be used to determine the parsimonious set of salient features. In a forward screening method, I ANNs are trained each containing one candidate input feature. One feature is retained and $I-1$ ANNs are trained each

83

containing the retained feature and one of the remaining candidate input features. The ANNs trained may use the same ANN or a new ANN. Another feature is retained and $I-2$ ANNs are trained each containing the retained features and one of the remaining candidate input features. And so on until all of the input features are retained. A total of $\frac{I \cdot (I+1)}{2}$ ANNs are trained in a forward screening method [84].

A MOE such as CA_{valid} may be used to determine the parsimonious set of salient features. Figure 14 shows the number of trained ANNs as a function of the number of candidate input features for both the backwards screening method and the forward screening method. The number of ANNs to train in forward screening methods increases exponentially as the number of candidate input features increases as seen in Figure 14.

In practical instances, the parsimonious set of salient features derived from the

Figure 14. Computational Efficiency of Backwards Screening versus Forward Screening
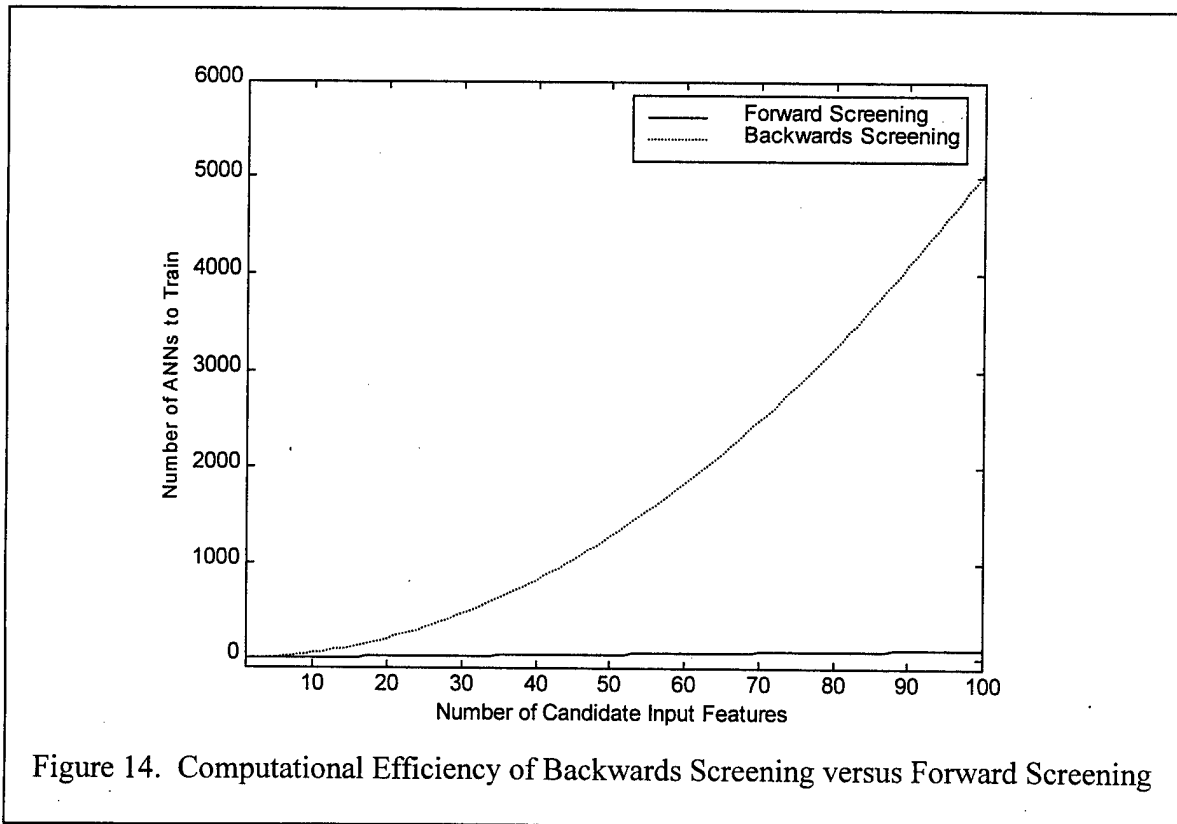
backwards screening method may not differ substantially from that derived from the

forward screening method [84].

4.1   Introduction

This chapter provides a literature review of *psychophysiological* approaches used to measure mental workload. Topics covered include a motivation for the importance of researching pilot workload and air traffic controller workload, physiological responses that measure psychological state, collecting and preprocessing EEG, challenges using EEG, EEG analysis techniques, and modeling mental workload using ANNs.

4.2   Motivation to Research Pilot Workload and Air Traffic Controller Workload

The issue of pilot workload is important to the USAF because pilot overload or task saturation is decreasing mission effectiveness and, in some extreme cases, causing loss of lives [3]. The modern aircraft is not an ideal work station for human operators. The fighter pilot must perform complex cognitive tasks while exposed to acceleration levels up to +9 Gs. Between 1986 and 1995, the USAF lost 14 of its fighter pilots to G-induced loss of consciousness (G-LOC) [3]. All but one of these 14 mishaps occurred during demanding portions of the flight under conditions of high workload [3]. These mishaps resulted from the pilot being so task saturated, that he failed to perform an adequate anti-G straining maneuver [3]. Some day, instrumentation may be in every cockpit to monitor a pilot's workload in order to warn a pilot that overload or task saturation is imminent. The ability to monitor a pilot's workload will also have far-reaching results in the research and development of future cockpits.

Like flying an airplane, air traffic control has long been regarded as a complex, demanding, and at times task saturating endeavor. In the past decade, air traffic control workload has become a particular concern of the Federal Aviation Administration (FAA). This is primarily due to the increased traffic load in our national airspace system. Reports from NASA's Aviation Safety Reporting System show that controller errors (i.e. monitoring failures, improperly executed handoffs, wrong heading assignments, or wrong altitude assignments) are associated most frequently with increases in workload factors such as traffic volume and frequency congestion [96]. The FAA's *National Airspace Plan* proposes to increase the automation of air traffic control in order to decrease air traffic controller workload and safely meet the ever rising demand for air traffic services [34]. Relevant measures must be developed in order to effectively show that automation is indeed lowering the workload of air traffic controllers [15, 177].

4.3   *Physiological Responses that Measure Psychological State*

There is a tremendous amount of research that has been conducted that shows that there are indeed consistent changes in several physiological responses that are related to the nature and level of mental activity. Measured physiological responses that are associated with psychological state are termed *psychophysiological* measures. Psychophysiological measures can be continuous, non-intrusive, and relatively easy to collect [169, 170, 175, 177]. Attempts to model mental activity using electrophysiological responses thus far have been initially successful and show much promise. Consistent measurements of physiological responses shown to be related to the nature and the intensity of mental activities are the following: EOG, ECG, respiration

87

gauges, electromyography (EMG), rheoencephalography (REG), blood pressure and blood flow, temperature, skin resistance and skin conductance, and EEG. EEG measurements, which show the most promise for analyzing mental activity can be divided into two types: EPs and ongoing activity.

4.3.1 Peripheral Psychophysiological Measures

Psychophysiological measures that do not directly measure the brain are termed *peripheral* psychophysiological features. Those showing the most promise are EOG, ECG, respiration measures, and REG. There are several other peripheral psychophysiological measures that are very briefly described. In the last few years, peripheral psychophysiological measures have been used to aid in monitoring pilot workload in addition to air traffic controller workload and include eye blink rate, heart rate, heart rate variability, and respiration rate [15, 169, 170, 175, 177].

4.3.1.1 Electro-oculography (EOG)

EOG features have been shown to reflect cognitive state by monitoring eye movement, eye blinks, direction of gaze, and eye closure [63]. Eye blink rate increases reflect the deterioration in attention and performance which occur over a prolonged task [8, 10]. An increase in eye blink rate and duration indicate fatigue or lack of vigilance [63]. As visual information processing demands increase, eye blink rate decreases reflecting the brain's attempt to not "miss anything." Eye blink rate decreases and latency increases as mental workload increases [7]. Eye blink durations increase with time on task [44]. Both eye position and pupil dilation appear to vary systematically with

88

mental workload [53]. The pupils dilate with increased task difficulty, increased mental workload, activation, and arousal [9, 21, 29]. Eye blink rate has been shown to be a sensitive measure to visual workload [15, 170, 173, 175, 177]. Eye blink rate typically decreases when visual demands increase [15, 170, 173, 175, 177]. The power of the EOG signal provides eye activity information by representing both eye blinks and eye movement. Eye movement typically increases as visual demands increase [144].

4.3.1.2 Electrocardiography (ECG)

ECG provides a measure of the cardiovascular activity in terms of rate per unit time and change in heart period across beats. Other possible measurements include interbeat interval, rate-of-change, maximum beat-to-beat periods, and minimum beat-to-beat periods. Heart rate has been shown to increase with stress [114] and activation [29]. The heart rate response to stimuli in a task environment is more often characterized by a complex pattern of deceleration and acceleration [63]. Lacey proposes that heart rate deceleration reflects a receptivity to external stimulation whereas acceleration occurs if the situation is found, after initial attention, to warrant an increase in energy reflexes [74]. Heart rate increases during periods of increased mental workload such as during take-offs and landings [55, 117, 118]. More consistent relationships with mental workload have been reported for heart rate variability. The general finding has been that with increased attention and mental workload, heart rate variability decreases [130, 156]. A frequently used technique to reveal mental workload effect is a spectral analysis of the beat-to-beat time interval with a focus on the power in the 0.1 Hertz (Hz) band [97]. Of particular interest has been the component of heart rate variability related to respiratory sinus

arrhythmia. The beat-to-beat regularity of the heart reflects mediation by the central nervous system. Porges has developed a moving polynomial filter technique that removes the slowly shifting baseline from the interbeat interval data over time in order to reveal the faster oscillations due to respiratory sinus arrhythmia. [108, 109]. Increased heart rate is typically associated with increased workload [15, 169, 170, 175, 177]. Also, the variability of the cardiac rhythm decreases with increased task difficulty [15, 169, 170, 175, 177]. Previous pilot workload research shows that heart interbeat interval decreases as workload increases during fighter aircraft air-to-ground missions [170].

4.3.1.3 Respiration Gauges

Principal measures from respiration gauges include respiration rate, interbreath interval, average volume, timing of respiration, inspiratory pause, expiration, expiratory pause, and the volume of air expired. Mercury-in-silastic tubing strain gauges measure the thorax and the abdomen. Thermostors mounted in an oxygen mask sense the volume of air expired. Williges and Wierwille give evidence that respiration becomes more shallow, regular, and rapid with increased mental workload [168]. Like heart rate, increased respiration rate is typically associated with increased workload [15, 169, 170, 175, 177]. Also, the variability of the breathing rhythm decreases with increased task difficulty [15, 169, 170, 175, 177]. Interbreath interval has been reported to decrease with higher levels of mental effort [20, 164, 165, 169].

4.3.1.4 Electromyography (EMG)

EMG recording from surface electrodes detect muscle tone or movement

mediated by selected muscle groups. The forehead and the masseter muscle (the large muscle used in chewing that raises the lower jaw) indicate overall tension levels [63]. Tension level typically averaged over 0.1 to 0.5 seconds is used to derive mean level, variance of the level, minimum level, maximum level, and number of increases over a threshold. Muscle tension increases with arousal, stress, and activation [29, 30]. Increased EMG activity is also associated with the onset of fatigue [63]. Several studies have reported relationships between increased EMG activity and increased mental workload or task difficulty [22, 67] but it is not clear how sensitive EMG is as an index to small changes in mental workload.

4.3.1.5 Rheoencephalography (REG)

REG provides measurement of cerebral circulation, cerebral neural activity, and intracranial blood flow [94]. REG measures can provide indices of left and right hemisphere hemodynamic changes. Increasing mental activity in a given brain area is highly correlated with increasing intracranial blood flow in the same brain area [94]. REG changes to mental activities have been described by Ingvar and Risberg [64], Jacquy et al. [66], Montgomery et al. [93, 95], and Piraux et al. [107].

4.3.1.6 Other Peripheral Psychophysiological Measures

There are many other psychophysiological measures than those described above. A few are mentioned here. Evoked magnetic fields have been correlated with attention and subjective probability [104]. In addition, blood pressure and blood flow provide useful information about cardiovascular status and complements the information

available from heart rate and heart rate variability. Core temperature monitored from a swallowed "pill" with telemetry and skin temperature changes have been related to mental workload [52]. Finally, skin resistance and skin conductance have some value for indicating changes in arousal and stress but have not yet shown utility for inferring cognitive states [63].

4.3.2 Electroencephalography (EEG)

As early as 1929, Hans Berger, the discoverer of EEG, asked:

> Will it be possible to demonstrate intellectual processes by means of the EEG? [13: 569].

The development of the link between EEG and mental activity began 50 years ago [43]. Early investigators immediately showed that mental activity profoundly effects scalp electricity [43]. Since the brain is the organ responsible for evaluating sensory information and then making and carrying out decisions based upon that sensory information, ongoing activity as measured by EEG would seem to hold a great deal of potential for measuring mental workload. Wilson states in the preface of the special issue of *Biological Psychology* on "EEG in Basic and Applied Setting":

> The EEG can be used to derive a more complete understanding of the workings of the human brain and also can be correlated with human performance to provide insights into cognition. [171: vii]

Peripheral psychophysiological measures have shown limited success in classifying mental workload. Gevins performed much of the early research in correlating EEG to mental workload [41, 42.] In the last few years, EEG measures have also been used to classify pilot workload in addition to air traffic controller workload [15, 143, 170, 175, 177]. Measures of the brain's electrical activity have only been recently added to

the arsenal of pilot workload measurements in work done by Caldwell et al. [17, 18, 19], Russell et al. [129], Sterman et al. [141, 142, 143] and Wilson et al. [175]. EEG has also only been recently added to the arsenal of air traffic controller workload measurements in work done by Brookings et al. [15] and Wilson et al. [177]. EEG currently appears to be our best "window to the brain." There are generally two types of EEG data used as features for determining a relationship with a human activity: EPs and ongoing EEG activity. EPs are typically the response of an EEG channel to some stimuli and consists of the amplitude of the EEG channel's signal.

4.3.2.1   Evoked Potentials (EP)

EPs have shown to vary reliably with cognitive processes [27], selective attention [60], expectancy [148], discrimination processes [112], and response preparation [159]. There is a body of research that has shown relationships between EPs and mental workload. There is evidence that EPs are sensitive to the cognitive processes affected by mental workload [98]. In almost all cases, averaged EPs are used. Single trial EPs still require much research.

In 1995, Skrandies used averaged EPs to analyze mental activity evoked by localized visual stimuli and by stereroscopic stimulation [134]. He showed changes in averaged EPs by a checkerboard reversing in contrast [134]. Skrandies also demonstrated changes in averaged EPs by a stereoscopic checkerboard pattern moving in depth [134]. In addition, Skrandies showed changes in averaged EPs during the time course of perceptual learning in human adults [134].

Ullsperger and Grune successfully used the P300 component of averaged EPs

during mental comparison of compound digits in 1995 [155]. The test subject was asked to decide whether a physically larger digit was numerically larger or smaller than a physically smaller one. The averaged P300 amplitude increased with increasing differences between the numbers compared [155].

Trejo et al. utilized EPs in 1995 to index performance in visual display-monitoring tasks. For signal detection and running memory, averaged P300 amplitudes increased when the task was engaged and was greater for accurate response trials then inaccurate response trials [154]. The P300 latency has been shown to vary with only a subset of manipulations that affect overt reaction time. This suggests that the timing of the P300 indexes the completion of a stimulus evaluation process, independent of the response selection process [86]. Steady-state EPs elicited by rapid, periodic stimulation by a checkerboard have also been reported to reflect mental workload when the checkerboard was presented concurrently with task performance [177].

Hohnsbein et al. showed in 1995 that increases in mental workload can induce acceleration or deceleration of specific processing stages which can be monitored or observed by latency changes of affiliated EPs [61]. Their approach established a relationship between EPs and information processing stages. The P300 component can show both stimulus evaluation and response selection [61]. Hohnsbein also demonstrated that as time pressures increase, the latency of the P300 component for response selection was shortened [61].

The results from a 1995 study on Navy radar operators by Kramer et al. also suggest that EPs are an effective method for evaluating increases in mental workload in complex tasks [73]. Amplitudes of the N100 and N200 components decreased during a

94

low-load radar monitoring task and then increased during high-load radar monitoring tasks [73]. The load of the radar monitoring task varied in the density and type of targets to be detected and identified.

4.3.2.2 Ongoing Electroencephalography (EEG)

It wasn't until the early 1990s that derived measures from ongoing EEG proved to be reliable indicators of mental workload. In 1995, Wilson and Fisher were able to classify with 86% accuracy which of 14 tasks each of seven subjects had performed using ongoing EEG activity [174]. The tasks used by Wilson and Fisher included auditory and visual stimuli, visual and auditory memory tasks, a spatial processing task, and a visual monitoring task. The memory, spatial, and monitoring tasks each had two levels of difficulty. The artifact-removed EEG signal was processed via a fast-Fourier transform (FFT) and divided up into frequency bands as summarized in Table 3. The spectrum power of each frequency band proved to be the most reliable measurement. This study by Wilson and Fisher demonstrated that EEG can be used to discriminate between human cognitive activity involving a number of different tasks [174].

Sterman et al. utilized ongoing EEG during signal detection, flight simulation, and actual flight performance [141, 142, 143]. Their study provided evidence that distinct

Table 3. Frequency Band Designations

Band	Symbol	Frequency
Delta	Δ	1.0 - 3.0 Hz
Theta	θ	4.0 - 7.0 Hz
Alpha	α	8.0 - 12.0 Hz
Beta	β	13.0 - 30.0 Hz
UltraBeta	$\mu\beta$	31.0 - 42.0 Hz

95

EEG frequency changes are related to psychomotor behavior, signal processing, and intrinsic attentional modulation during complex performance. The research by Sterman et al. indicated that EEG can provide a valid and objective index for mental effort and, in addition, may reveal task-related cognitive resource allocation, task mastery, and task overload [141, 142, 143].

Gundel et al. showed a direct relationship in 1995 between levels of sleepiness and ongoing EEG activity in airline pilots [51]. Their study utilized broadband power in four bands (Δ, θ, α, and β), α desynchronization index, and α peak frequency [51].

Horst ranks EEG as the top feature for demonstrating a relationship with mental activities as in the following circumstances:

> Changes in the predominant frequencies in the EEG with levels of arousal and activation have been known for some time [80, 83]. An alert person performing an engaging task shows predominately low amplitude, fast frequency β (13-30 Hz) activity. An awake, but less alert, person shows an increased incidence of high amplitude α (8-12 Hz) activity. With the onset of drowsiness, slower frequency θ (4-7 Hz) activity enters the spectrum and in the early stages of sleep, very high amplitude, slow frequency Δ (1-3 Hz) waves dominate. The generalized effect of stress, activation, or arousal is, therefore, a shift towards the faster frequencies, often with an abrupt blocking of the α rhythm [29, 83]. Fatigue and boredom generally shift the spectrum in the other direction towards the lower frequencies. [63: 31]

John and Easton's 1995 study investigated both ongoing EEG activity and EPs [68]. John and Easton used "tracer strategy" to show that different levels of mental workload such as easy, moderately difficult, and extremely demanding can be statistically monitored using power in the EEG spectrum and EP waveshapes [68]. Their method was applied during performance of an audio-visual continuous pursuit task in which the target and pursuer were labeled at different frequencies and during performance of a delayed

match from sample tasks in which sets of letters, numbers, or faces modulated at a specific frequency had to be retrieved from working memory [68].

Past investigations show psychophysiological features derived from EEG as good indicators of mental workload [15, 41, 42, 46, 47, 49, 129, 143, 170, 174, 177].

4.4   Collecting and Preprocessing Electroencephalography (EEG)

In this research, EEG from as little as six channels up to 128 channels can be collected from electrode sites located on the head using the Workload Assessment Monitor (WAM) [172]. One or two electrodes typically serve as reference. The location of the sites are based on the International 10-20 system. Figure 15 and Figure 16 show the locations of six electrode sites used in this research. The electrodes in Figure 15 are

Figure 15. Location of Six EEG Electrodes Viewed From Top of Head
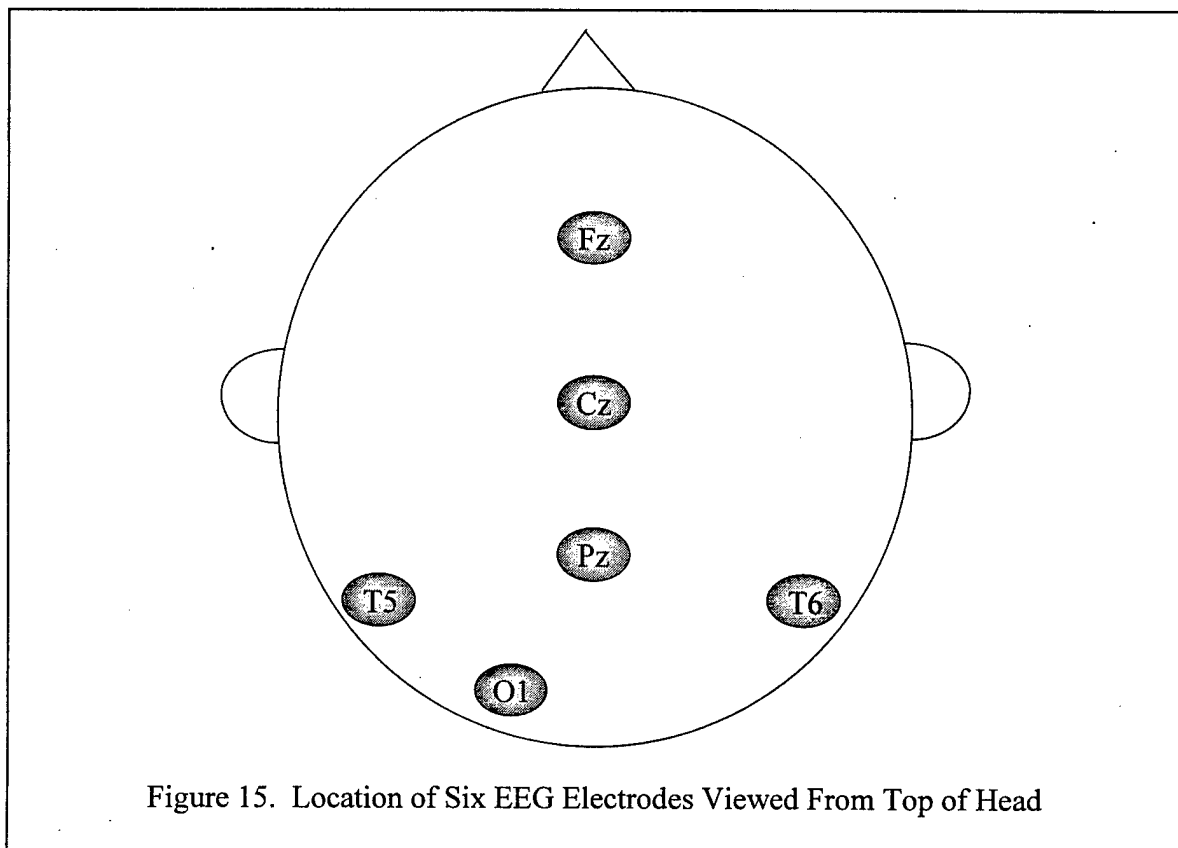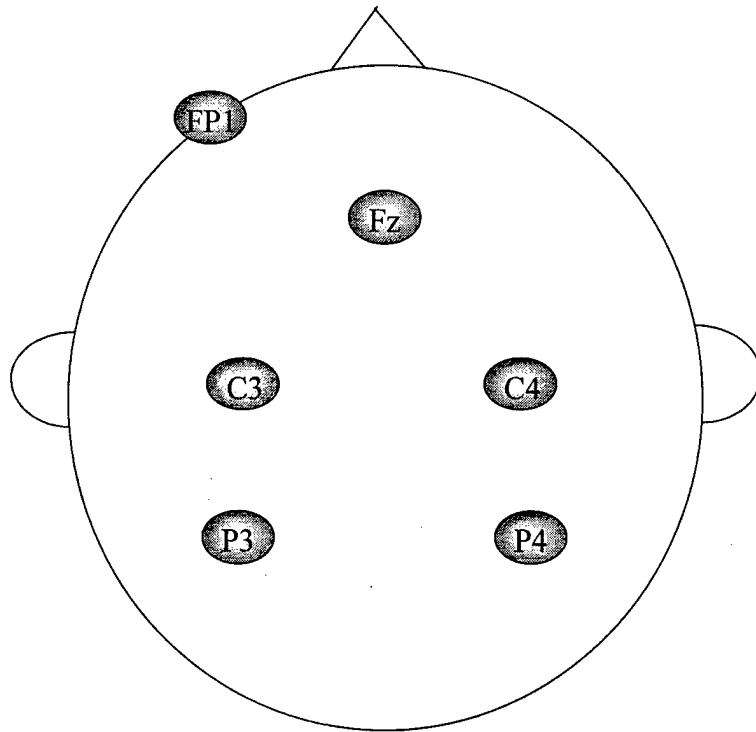
Figure 16. Alternate Location of Six EEG Electrodes Viewed From Top of Head

Fz, Cz, Pz, T5, T6, and O1. The electrodes in Figure 16 are FP1, Fz, C3, C4, P3, and P4.

EEG data for classifying pilot workload and air traffic controller workload is typically sampled by the WAM at 128 Hz [129]. The WAM preprocesses the EEG signals. First, the EEG data is amplified 60,000 times using bioamplifiers with filter cutoff settings at 1.0 Hz and 40.0 Hz [129]. Then, the WAM removes all artifacts caused by eye blinks [172].

The preprocessed data from the WAM is then further processed using *MATLAB* code. One way of processing the EEG is to take a 64-point, 1-Hz resolution FFT every second [129]. Then, the power spectral density is calculated using the direct method of a periodogram [129]. A 10-second window with 50% overlap is applied for smoothing processes (prior pilot workload classification methods, including discriminant analysis

and Bayes quadratic classification, indicate that this results in the highest workload classification accuracies) [129]. The periodogram is then grouped into frequency bands as listed in Table 3 or in more detail as listed in Table 4.

Both the logarithm (log) of the average power and the variance of the power over each window for each frequency band for each electrode may be calculated to provide features [129]. The log of the average power and the variance of the power over each window have thus far been good features of EEG for classifying pilot workload [129].

4.5   Challenges Using Electroencephalography (EEG)

EEG shows the most promise of being sensitive to the levels or intensity of mental activity such as mental workload. The use of EEG for classifying types of mental activities and for classifying the levels of these mental activities is only in its infancy. As such, there are challenges facing the use of EEG for measuring mental workload.

Table 4. Alternate Frequency Band Designations

Band	Symbol	Frequency
Delta	Δ	1.0 - 3.0 Hz
Theta	θ	4.0 - 7.0 Hz
Alpha	α	8.0 - 11.0 Hz
Alpha1	α_1	8.0 - 9.0 Hz
Alpha2	α_2	10.0 - 11.0 Hz
Beta1	β_1	12.0 - 14.0 Hz
Beta2	β_2	15.0 - 30.0 Hz
UltraBeta1	$\mu\beta_1$	31.0 - 36.0 Hz
UltraBeta2	$\mu\beta_2$	37.0 - 42.0 Hz

4.5.1 Evoked Potentials (EP)

A major issue with using EPs is that since EPs simply use the amplitude of the EEG signal, single trial EP analysis is difficult to do because the signal-to-noise ratio (SNR) between a typical EP and the ongoing EEG activity is around -20 decibels (dB). As such, experiments are set up to average 30 to 50 trials of the EP. Problems with this include synchronizing the beginning and end points of the stimuli. Other problems develop when the length of the stimuli differs.

4.5.2 Ongoing Electroencephalography (EEG)

This research will use ongoing EEG. There are several challenges, though, to using ongoing EEG.

4.5.2.1 Nonstationarity

The EEG signal is very complex. It varies in time and is thus typically nonstationary. This may cause problems with some analysis methods. Attempts have been made to make EEG quasistationary usually by subdividing the EEG into "epochs" with the same statistical properties. (Note to reader: Do not to confuse the term epoch with the same typically used in ANNs). In a study conducted by Isaksson and Wennberg, some 90% of the EEG signals investigated had time-invariant properties after 20 seconds, whereas less than 75% remained time-invariant after 60 seconds [65].

Empirical observations indicate that EEG obtained under equivalent behavioral conditions show highly stable characteristics [100]. Since behavioral conditions can change in a very short period of time, it is safe to assume that relatively short period EEG

time intervals around 10 seconds recorded under constant behavioral conditions are quasistationary [100]. Elul remarked that EEG is related to the intermittent changes in the synchrony of cortical neurons and so EEG should be characterized as a series of short time periods instead of as a continuous process [32]. To remedy this problem, all ongoing EEG used in this dissertation will be averaged over a 10-second moving window with 50% overlap as already mentioned in Section 4.4.

4.5.2.2 Cross Correlation

Not only is EEG typically nonstationary, but adjacent samples of the EEG are usually highly correlated. As such, consecutive samples of EEG are usually not independent. In the space dimension, the EEG is dependent on the location of the electrodes on the head. The montage of the electrodes must be carefully selected so as to reflect necessary topographic characteristics. Decisions as to the number of electrodes must also be decided. For this research, only six electrodes will be used to account for these shortcoming.

4.5.2.3 Consistency

Many factors may effect an individual's EEG: sex, age, medication, sleep, coffee, food, et cetera. An individual's ongoing EEG response to a stimuli may change day-to-day. It may even change within a day depending upon the person's behavioral state or when he had his last cup of coffee. There is also no assurance that the EEG of individual *A* may be the same as individual *B*. This research will take these drawbacks into

consideration where possible. These consistency issues, though, are limitations to this dissertation.

4.5.2.4 Relationship between Electroencephalography (EEG) and Human Mental Activity

Problems may exist in attempting to correlate EEG features to some component of human mental activity. Many statistical correlation techniques assume a *linear* relationship. This may not be a valid assumption, especially when mother nature and the human body is involved. As such, nonlinear techniques using feature saliency measures in ANNs (see Section 3.4) are utilized in this dissertation. There also may be serious cross-correlation between features, especially among channels that are spatially close to each other. Though feature saliency measures do not directly account for cross-correlation, this research will use saliency screening methods in ANNs (see Section 3.5) for determining parsimonious sets of salient features. These methods may account for cross-correlation.

4.5.2.5 Quantification

In order to discover, in the EEG, data relevant to some component of human mental activity, quantification of the EEG signal must take place. Classical quantification of the EEG signal involves measuring frequency and / or amplitude [100]. Major problems to finding the relationship between EEG and human mental activity hinges on whether EEG signals change in relationship to human mental activity. It also depends upon how synchronous the EEG signals really are to different derivations of

human mental activity. Quantification of the EEG attempts to describe numerically the EEG phenomena of spikes, sharp waves, and other abnormal patterns. EEG is a stochastic process with some measurable statistical measures like average amplitude and average frequency. The EEG may be characterized by its probability distribution, its moments (i.e. mean, variance, skewness, and kurtosis), its frequency spectra, or by its correlation function. For this research, ANNs will be used for modeling the EEG since ANNs do not make any assumptions about the functional form of the underlying population density distribution of in the input features [160].

4.5.2.6 Sampling Frequency

Selection of a sampling frequency is key to the success of quantifying EEG. Equidistant time intervals are highly recommended. The choice of a sampling frequency is typically based upon *Nyquist's sampling theory* which states that the sampling frequency must be at least equal to $2 \cdot f_N$ where f_N is the folding or Nyquist frequency assuming that the EEG signal denoted as $x(t)$ has a frequency spectrum denoted as $X(f)$ such that $X(f) = 0$ for f_N. The *sampling theorem* forces the use of a low-pass filter to ensure that all frequency components greater than f_N are removed. If neurocognitive relationships are sought after, a very small sampling frequency must be used since some cognitive functions change in a fraction of a second. To account for proper sampling, this research will sample the EEG at 128 Hz as detailed in Section 4.4.

4.6 Previous Electroencephalography (EEG) Analysis Methods

There are many EEG analysis methods that can be utilized to possibly discover

EEG data relevant to some component of human activity. EEG analysis methods can be divided into five groups:

1. Nonparametric methods
2. Parametric methods
3. Mimetic analysis
4. Matched filtering or template matching
5. Topographic analysis [100].

These analysis methods, their advantages and disadvantages, and their assumptions are briefly described.

4.6.1 Nonparametric Methods

The majority of the methods for analyzing EEG are nonparametric.

4.6.1.1 Amplitude Distributions

An EEG signal can be characterized by its amplitude distribution and moments. The first question typically asked is whether the amplitude distribution is normal. There have been several studies that determined the appropriate sampling rates and epoch lengths in order to properly assume that the EEG amplitude distributions are normal. However, all of these studies violated either the independence or stationary requirement of goodness-of-fit tests. So the question still remains as to whether EEG is a Gaussian phenomenon.

4.6.1.2 Interval Analysis

Interval analysis is a very simple method to analyze EEG signals that has found success in quantifying EEG changes induced by psychoactive drugs, in monitoring EEG changes during anesthesia, in psychiatry, and in sleep research [100]. Interval analysis

computes the zero-crossings of the original EEG signal along with those of the signal's first and second derivatives. Disadvantages to interval analysis include its underestimation of the contribution of low frequency components and its overestimation of fast frequency components. Another disadvantage of interval analysis is its sensitivity to high frequency noise. This problem of high frequency noise sensitivity can be alleviated by creating a dead band so that no zero-crossings can be detected when the signal has an amplitude between those limits. The major advantage of internal analysis is ease of use.

4.6.1.3 Interval-Amplitude Scatter Plots

Interval-amplitude analysis decomposes the EEG into waves or half-waves, defined both in time, by the interval between zero-crossings, and in amplitude by the peak-to-trough amplitudes. This requires a minimum sampling rate of 250 Hz and a dead band to avoid the influence of high frequency noise.

4.6.1.4 Correlation Analysis

Correlation analysis was used extensively in the 1950s and 1960s. It is the forerunner to today's spectral analysis. Computing the correlations were time consuming and thus, this method was not widely used. A simplified correlation function called the "polarity coincidence correlation function" replaced the signals $x(t)$ and $x(t-lag)$ for $t = 1,2,...T$ and for $lag = 1,2,...L$ by their signs. Another form of simplified correlation is called "auto-averaging" and consists of making pulses at a certain phase of the EEG (i.e. zero-crossing, peak, and trough) that are then used to trigger a device that averages

105

the same signal (auto-averaging) or another signal (cross-averaging). This allowed for the detection of rhythmic EEG. However, correlation analysis is difficult when more than one dominant rhythm is present. The correlation analysis method has lost its popularity with the advent of the FFT.

4.6.1.5 Complex Demodulation

Complex demodulation allows for a particular frequency component such as 10 Hz to be detected and followed as a function of time. However, a priori knowledge of the component is necessary. An "analysis oscillator" at the given frequency (in this case, 10 Hz) is set and the oscillator output and the EEG are then multiplied. The product contains components at the sum (20 Hz) and at the difference (0 Hz). The product is smoothed so that only the difference (0 Hz) is considered. Now, phase and amplitude of EEG frequency components can be detected and their modulation in time determined. Complex demodulation has been successfully used to analyze rhythmic components of visual EPs and sleep spindles. This method is very similar to Fourier analysis.

4.6.1.6 Power Spectra Analysis

Power spectra analysis provides the most appropriate method of EEG analysis. Analogue filtering was used to decompose EEG signals into frequency components until the 1960s when the FFT was developed. A spectral window, defined by its form and duration, must be selected. Using a window with a large base reduces the variance but increases the bias. An excessively large window greatly decreases the frequency resolution. The selection of the spectral window depends upon the practical use of the

spectral analysis. For typical clinical use, it is common to compute average spectra by making averages of ensembles of 10 epochs of 10 seconds using an elliptic window five samples wide for smoothing [100].

There are many ways to plot power spectra. The log of the power intensity is typically used instead of power intensity because confidence intervals of the log power intensity are independent of the spectral intensity. The square root of the spectral intensity may also be used. If attention is to be placed on the lower frequencies (Δ and θ), it is highly recommended to compress the frequency scale in the higher frequencies by plotting log Hz.

4.6.1.7 Time-Varying Spectra

Time-varying spectra are computed to analyze slowly changing EEG and is useful for an overall view of EEG spectral changes for intraoperative or sleep monitoring [100]. Contour plots may provide an interpretable visual display of the evolution of power spectra as a function of time. Time-varying power spectra is particularly helpful when trying to characterize EEG changes in relation to a specific event such as eyes closing, eyes opening, and word association tests [100]. The difficulty here lies in quantifying time-locked changes in EEG spectra by ensemble averaging.

Since baseline EEG (i.e. pre-event segment) can change trial to trial, statistical analysis based on ensemble averages and standard deviations can be at times difficult. Because the baseline values may vary dramatically, one runs the risk of failing to detect real EEG changes related to a particular event if one compares only mean values. One

proposal to alleviate this problem is to analyze EEG epochs immediately before and immediately after the event causing the change.

Statistical evaluation of spectra is possible to determine if two sets of EEG power spectra differ significantly. The sets might have been obtained under two different behavior conditions (i.e. placebo versus psychotropic drug). It is necessary to state that the power spectrum of EEG is an estimate and such, it has variance. Analysis of variance (ANOVA) and t-tests are often conducted. Since the normality of EEG is not known, though, nonparametric tests such as the Wilcoxon or Mann-Whitney may be more applicable.

4.6.1.8 Cross-Spectral Analysis

Cross-spectral analysis is an important part of EEG spectral analysis because it allows for the quantification of the relationship between different EEG signals. Cross-power spectrum is the product of the smoothed FFT of one signal and the complex conjugate of another. A so-called coherence function is then computed and normalized. The coherence function has been used in several investigations of EEG signal generation and their relation to brain functions, including studies of hippocampal θ band rhythms, on limbic structures in humans, on thalamic and cortical α band rhythms, on sleep stages in humans, and on EEG development in babies [100]. The major disadvantage to the coherence function is its assumption of a *linear* relationship between the two EEG signals.

The use of coherence functions brings up some interesting points. For example, is it possible to differentiate spectral components with frequencies lying close to each

other? In the case of α and μ band rhythms (the μ band frequency, though not listed in Table 3 or Table 4, is slightly higher than the α band frequency) it is near impossible to differentiate between the two in power spectra plots. Yet, α and μ band rhythms are easily separated using coherence functions. As such, coherence may show promise for determining topographical relations.

A so-called phase function can also be developed by cross-spectral analysis. A phase function provides information on the temporal relationship between two EEG signals. A time delay between two signals can be concluded with certainty only if there is a *linear* relationship between phase and frequency within a certain frequency band.

4.6.1.9 Bispectral Analysis

The main problem with using the power spectrum is that it assumes a stationary Gaussian process. Bispectra analysis allows for a way to analyze second-order spectra called the bispectrum. The bispectrum has not been used much except for showing a relationship between harmonic frequency components.

4.6.2 Parametric Models

No one yet knows if models of the biophyscial processes underlying the generation of EEG are more appropriate. There are a good number of parametric biophysical models to describe α band rhythm generation. Most utilize a filter network with parameters related to physiologically acceptable variables. Parametric models have thus far been successful in detecting epileptiform spikes and sharp waves [100].

4.6.2.1 Autoregressive Model

An autoregressive model can describe an EEG signal using just a few coefficients and allows for understanding of the time-varying properties of EEG. The autoregressive model is viewed as a filter described by a linear difference equation. The autoregressive model can also be used in an inverted way called the "inverse autoregressive filtering operation." Assuming the EEG signal is stationary, it is possible to approximate the EEG signal as filtered noise with a normal distribution. This inverse autoregressive filter then allows for the detection of nonstationarities in an EEG signal and has been highly successful in the detection of EEG transients of epileptics [100].

4.6.2.2 Kalman Filter

A Kalman filter allows for the analysis of time-varying signals. The input signal to the hypothetical processor responsible for generating the EEG signal is assumed to be normally distributed noise. A model is assumed in order to represent the observed signal and the process dynamics are represented by an autoregressive model. The Kalman filter requires a recursive algorithm that can up updated. The Kalman filter is not simple to implement because it is difficult to select the order of the model and the initial conditions.

4.6.2.3 Segmentation Analysis

Segmentation analysis provides a way to find those segments in an EEG signal that have unvarying statistical properties. Each segment is thus quasistationary and of varying length. An autoregressive model is used to create each segment. The major disadvantage to segmentation analysis is the difficulty of defining clinical-

neurophysiological boundaries between segments. It may, though, prove useful in reducing data of very long EEG signals.

4.6.3   Mimetic Analysis

Mimetic analysis is based on the general idea that automatic EEG analysis should mirror the visual analysis performed by an electroencephalographer in his daily practices. This analysis method simply uses tools common to the other methods already described, particularly interval-amplitude analysis. This method tends to overemphasize the high frequency components.

4.6.4   Matched Filtering or Template Matching

Matched filtering or template matching uses the cross-correlation between the EEG signal and some a priori defined pattern. This requires that the signal be aligned perfectly with the pattern. This also requires a "correct" template. This does not allow for time-variance. For example, what if the template lasts 0.3 seconds but yet the current signal of interest to match it up against lasts 0.35 seconds?

4.6.5   Topographical Analysis

Topographic analysis shows much promise. Topographical analysis may allow for understanding the distribution within the skull of the generators responsible for EEG. The biggest problem for topographic analysis is the inter-electrode distance. This problem is analogous to that of sampling frequency in the time domain (see Section 4.5.2.6). Aliasing error is used to determine if the representation of the electrical

111

potential distribution is good. If the ordinary 10-20 electrode system is used with an inter-electrode distance of 4.9 cm, the aliasing error is 6%. If the distance is reduced to 3.2 cm, the aliasing error is only 1%.

The total number of EEG channels for topographical analysis is typically either 64 or 128 channels depending upon the application. The first topographic maps called toposcopic displays modulated a series of light sources. The EEG signals were "visualized" in a multichannel oscilloscope and filmed. Spatial information was attained by watching the oscilloscope or film display. These topographic maps were difficult to interpret due to their complexity and variability in both time and space. Now two-dimensional contour plots are used. Interpolation (linear or not) is used to estimate the potentials between recording sites. Two-dimensional plots of averaged signals emphasize related EEG activity.

The appearance of the topographic maps depends on the way the EEG signals are recorded (bipolar, against a common reference electrode, or against the arithmetic average of all electrodes) and inter-electrode distances. Some studies now even have movies of topographic maps. With the advent of color video technology, topographic maps now exist for imaging power spectrum and EPs.

Topographic maps have been used extensively in the last few years. Topographic maps are widely used to assess brain function in patients suffering from brain ischemia, a form of obstruction of the blood supply in the brain [100]. Topographic maps have also been successful in displaying the regional distribution of reactivity in different frequency bands during auditory and visual stimulation [100].

Topographical maps allow for the analysis of the relationships between regions of the brain. Gevins used topographical analysis to study electrophysiological correlates of cognitive functions to determine the degree of interdependence between brain regions [100].

4.7   Modeling Mental Workload Using Artificial Neural Networks (ANN)

Previous research to classify pilot workload has used ANNs and, in particular, feedforward MLP ANNs as shown in Figure 3. The inputs to feedforward MLP ANNs used to classify workload may include peripheral psychophysiological features such as number of eye blinks, heart rate, heart interbeat interval, breathing rate, and respiration interbreath interval. In addition, the inputs may include features preprocessed in a variety of ways from EEG.

In 1996, Russell et al. classified the mental workload of five test pilot subjects using number of eye blinks, heart rate, heart interbeat interval, breathing rate, respiration interbreath interval, and ongoing EEG activity [129]. The mental workload was classified as low or high via a feedforward MLP ANN using approximately 130 input features. Artifact-removed EEG was preprocessed by a FFT and then split up into nine frequency bands as summarized in Table 4. EEG features were developed using the average log of power in addition to the variance of the power of the frequency bands. The classification accuracy for five test pilot was 83% [129].

ANNs show promise for classifying workload using EEG data due to the nonlinearity of EEG data, the generalization capabilities of ANNs, and the classification capabilities of ANNs. In particular, TDNNs and RNNs show promise for classifying

mental workload due to the temporal nature of EEG and the other psychophysiological

measures discussed.

5 Feasibility Studies Using Time Delay Neural Networks (TDNN) and Recurrent Neural Networks (RNN) to Classify Mental Workload

5.1 Introduction

This chapter provides a summary of two feasibility studies that were conducted. The two most common measures of the brain's electrical activity are EPs and ongoing activity of the EEG. Previous to this work, TDNNs and RNNs had never been used to classify EPs, ongoing EEG, or mental workload. The first study investigated the feasibility of using a TDNN to detect EPs in an EEG signal. The second study investigated the feasibility of using an Elman RNN to classify mental workload using ongoing EEG activity in the presence of noise.

5.2 Feasibility of Using Time Delay Neural Networks (TDNN) to Classify Evoked Potentials (EP)

5.2.1 Introduction

Whereas ongoing EEG focuses on a continuous recording of spontaneous brain electrical activity, the analysis of EPs focuses on segments of EEG activity that are related to (or evoked by) specific stimulus events. EPs are the small changes in voltage in EEG that are time locked to a stimulus or cognitive event. Two averaged EPs and several components (N1, P2, N2, and P3) are shown in Figure 17 [175]. The EPs in Figure 17 were collected from a USAF F-4 pilot performing the *oddball paradigm* in which two different tones were monaurally delivered to the pilot via a small speaker

placed inside his helmet ear cup [175]. The trials were repeated 100 times and the pilot was instructed to covertly count the number of times one of the tones was presented and then report this number at the end of the test [175]. The EP at the top of Figure 17 was collected during a baseline condition while the pilot was performing the oddball paradigm only. The other EP was collected while the pilot was flying low level and performing the oddball paradigm.

Analysis of EPs is predominantly accomplished using averaged EPs. Averaging is performed because the EEG is much larger in amplitude than the EP (especially at higher EEG frequencies). In fact, the SNR between an EP and EEG is typically around -20 dB. The maximum amplitude of an EP is usually around two microvolts and the maximum amplitude of EEG is usually around 20 microvolts. A large number of EPs are averaged in order to make distinctions between sizes and latencies of the EPs. The averaged EPs are then scored in terms of latencies (in milliseconds) and amplitudes (in microvolts) of each of several components (i.e. N1, P2, N2, and P3).

Most attempts at single EP analysis use a template derived from the averaged EP but have thus far been unreliable [20]. Only recently have investigators begun to focus on single EP responses. The major problem is determining what portion of the EEG signal is evoked by the response to the stimulus and what portion represents the continuation of ongoing background EEG. Unfortunately, background EEG typically looks like noise. If EPs are ever to be used to classify pilot workload or air traffic controller workload, then the ability to detect and then classify single EPs must be possible. The purpose of this feasibility study was to investigate the use of TDNNs to detect single EP responses in an EEG signal. This feasibility study has two parts.

116

Figure 17. Averaged EP Collected from F-4 Pilot Performing Oddball Paradigm
(Adapted from 175 with Permission from Dr. G.F. Wilson)
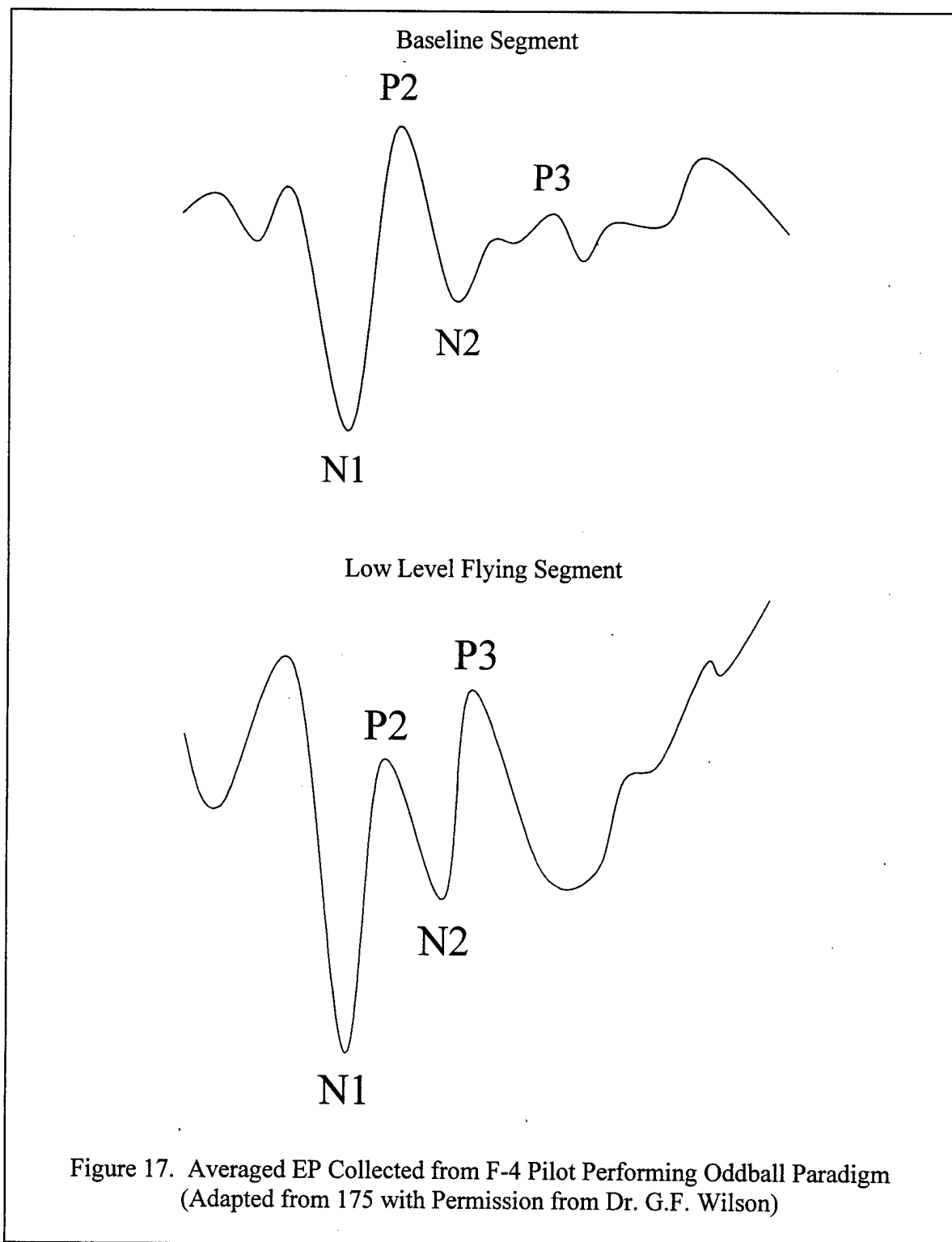
The first part attempted to detect a rectangle pulse in EEG at five varying SNRs. The second part attempted to detect an EP in EEG at five varying SNRs.

5.2.2 Data

In order to perform EP analysis, a sampling rate of 500 Hz is typically used on the collected EEG signals. This required sampling rate is considerable higher than the sampling rate of 128 Hz which is typically employed for EEG analysis when no EP analysis is necessary. Since this was only a feasibility study, smaller sampling rates were used.

5.2.2.1 Rectangle Pulse

An EEG signal was generated at a sampling rate of 50 Hz by summing five incommensurate sine waves using the following equation:

$$x(t) = 9.5 \cdot \sin\left(2 \cdot \pi \cdot 1.2 \cdot \frac{t}{50} + 1\right) + 4.4 \cdot \sin\left(2 \cdot \pi \cdot 4.1 \cdot \frac{t}{50}\right) +$$
$$3.4 \cdot \sin\left(2 \cdot \pi \cdot 12.9 \cdot \frac{t}{50} + 3\right) + 2.8 \cdot \sin\left(2 \cdot \pi \cdot 16.9 \cdot \frac{t}{50} + 2\right) + \qquad (93)$$
$$1.8 \cdot \sin\left(2 \cdot \pi \cdot 17.5 \cdot \frac{t}{50}\right)$$

The development of Equation 93 was based, in part, on actual EEG data using Table 5. The third column in Table 5 shows the number of sine waves used to represent each band

Table 5. Generated EEG Signal

Band	Frequency	Number of Sine Waves	Amplitude Ranking
Δ	1.0 - 3.0 Hz	1	1
θ	4.0 - 7.0 Hz	1	2
β_1	12.0 - 14.0 Hz	1	3
β_2	15.0 - 30.0 Hz	2	4, 5

118

in the development of the simulated EEG signal. For example, there is one sine wave to represent the Δ band, one sine wave to represent the θ band, one sine wave to represent the β_1 band, and two sine waves to represent the β_2 band. The fourth column in the table shows a rank ordering of the magnitude of the amplitude for each band. Thus, the sine wave corresponding to the Δ band has the highest relative amplitude, the sine wave corresponding to the θ band has the second highest relative amplitude, and so on. A total of 1000 samples representing 20 seconds were created such that the maximum peak amplitude was less than 20 microvolts. A plot of approximately two seconds of the EEG generated for the first part of this feasibility study is shown in Figure 18.

Next, a rectangle pulse was created to last 0.2 seconds which is the typical amount of time that an EP takes place. Since the sampling rate is 50 Hz, this equates to 10

Figure 18. Generated EEG Signal at 50 Hz
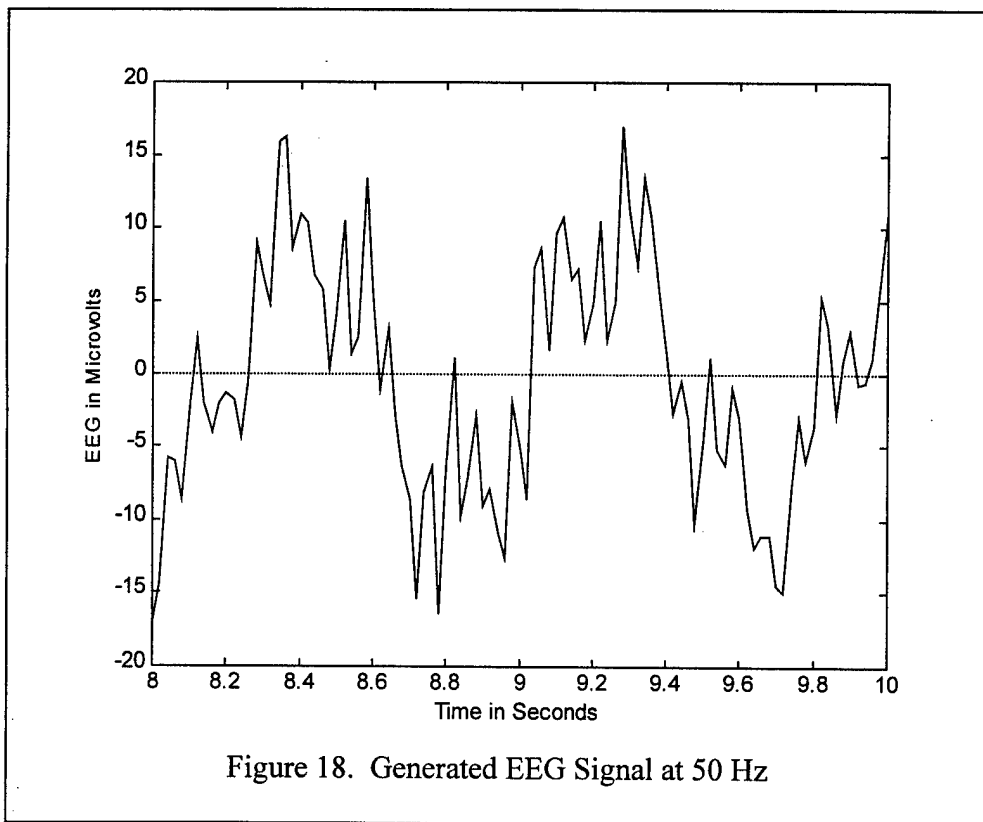
samples. In order to randomly place the rectangle pulse throughout the EEG signal, a random number generator was used to create 25 random start times between 1 and 1000. A start time was thrown out if it fell within 20 time samples of another start time. This resulted in a total of 17 rectangle pulses randomly placed throughout the EEG signal. A plot of approximately two seconds of the rectangle pulse can be seen in Figure 19. Five time series as shown in Figure 20 were created such that the SNR between the rectangle pulse and the EEG were different. The plots in Figure 20 are of interest to look at because the plots show that the human eye/brain can not detect the rectangle pulse when the SNR ≤ -2.59 dB. The SNR between the rectangle pulse and the EEG for each of the five time series was calculated following:

$$SNR = 20 \cdot \log\left(\frac{I(Pulse)}{I(EEG)}\right) \tag{94}$$

Figure 19. Generated Rectangle Pulse at 50 Hz

120
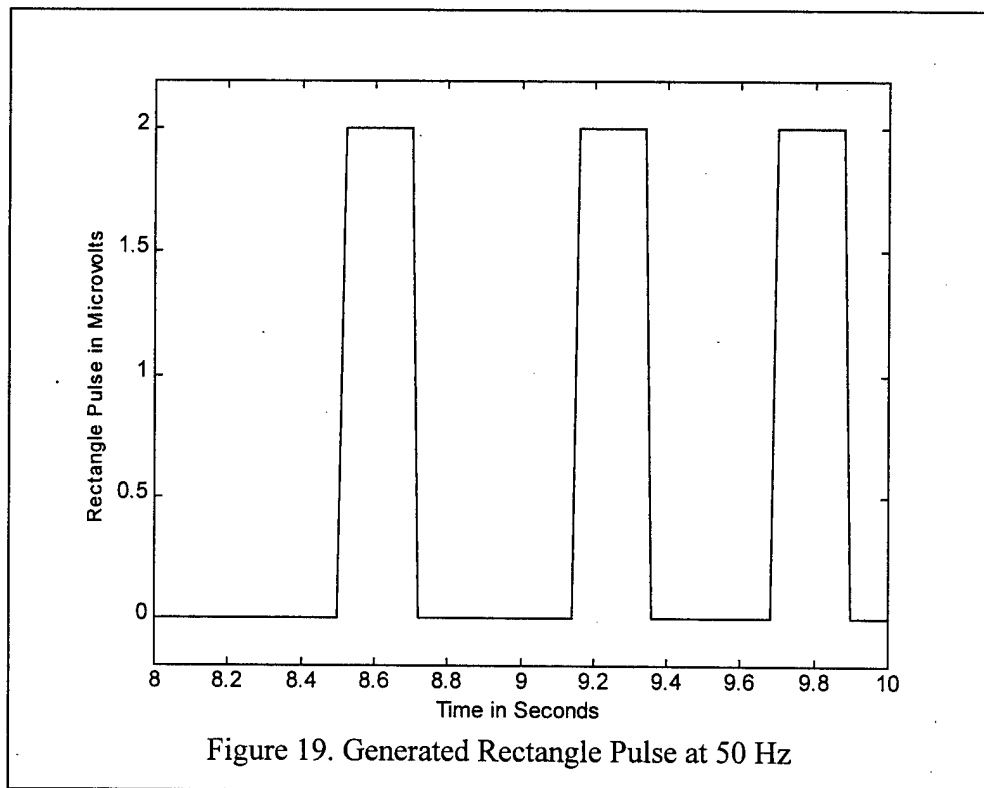
where $I(Pulse)$ is the *effective value* of the rectangle pulse and $I(EEG)$ is the effective

value of the EEG signal [70]. The effective value of a signal denoted as $I(S)$ is typically

computed [70] so that:
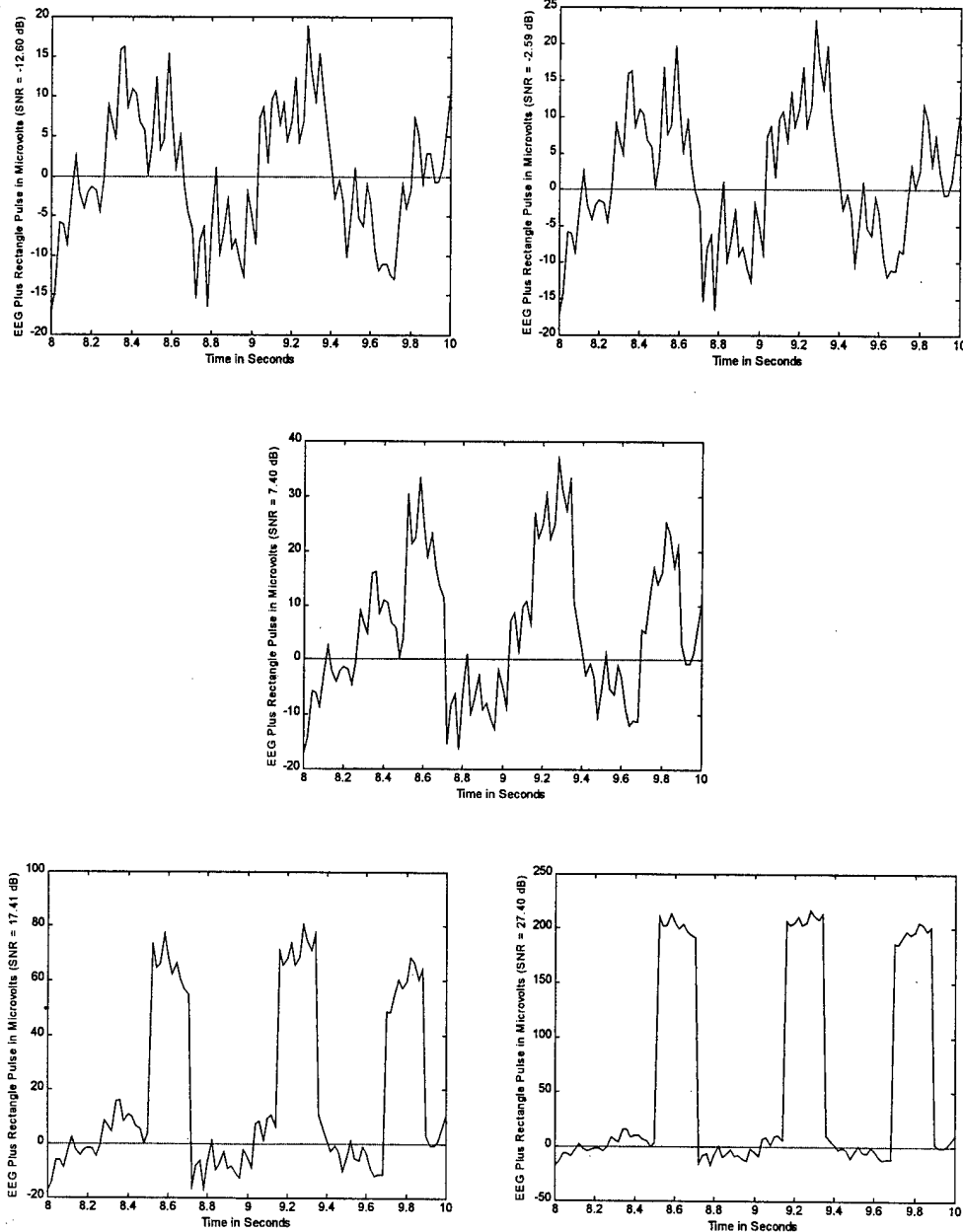
Figure 20. Generated EEG Signal with Generated Rectangle Pulse at Varying SNRs

$$I(S) = \sqrt{\frac{1}{T} \cdot \int_0^T [S(t)]^2 \, dt}$$ (95)

where $S(t)$ is the signal at time t and T is the total length of time the signal S took place. The effective value of the rectangle pulse was approximated using the trapezoidal rule [4] as:

$$I(Pulse) \approx \sqrt{\frac{1}{11} \cdot \left[\frac{1}{2} \cdot [\max(amp)]^2 + 9 \cdot [\max(amp)]^2 + \frac{1}{2} \cdot [\max(amp)]^2 \right]}$$ (96)

where $\max(amp)$ is the maximum amplitude. Since the EEG signal was the sum of five incommensurate sine waves, the effective value of the EEG signal was calculated [70] as:

$$I(EEG) = \sqrt{\frac{\sum_{w=1}^{S} \left[\max\left(amp_{\sin(w)}\right) \right]^2}{2}}$$ (97)

$$I(EEG) = \sqrt{\frac{9.5^2 + 4.4^2 + 3.4^2 + 2.8^2 + 1.8^2}{2}} = 8.13$$

where $\max\left(amp_{\sin(w)}\right)$ is the maximum amplitude of sine wave $w = 1,2,\ldots,5$. Table 6 summarizes the effective values of the rectangle pulse and EEG signal in addition to the SNR between the rectangle pulse and the EEG signal. The time series were divided into four classes as follows:

Table 6. Effective Value of Rectangle Pulse and EEG

Rectangle Max Amplitude	$I(Pulse)$	EEG Max Amplitude	$I(EEG)$	SNR
200.00	180.91	20.0	8.13	+27.40 dB
63.30	57.26	20.0	8.13	+17.41 dB
20.00	18.10	20.0	8.13	+7.40 dB
6.33	5.74	20.0	8.13	-2.59 dB
2.00	1.86	20.0	8.13	-12.60 dB

122

1. EEG only
2. Slight chance that an EP is present
3. EP is more than likely present
4. EP present

Class 1 contained all time samples that consisted of only EEG (no rectangle pulse) in addition to the first two time samples of the rectangle pulse: $Pulse_1$ and $Pulse_2$. Class 2 contained the next three time samples of the rectangle pulse: $Pulse_3$, $Pulse_4$, and $Pulse_5$. Class 3 contained the next three time samples of the rectangle pulse: $Pulse_6$, $Pulse_7$, and $Pulse_8$. Finally, class 4 contained the last two time samples of the rectangle pulse: $Pulse_9$ and $Pulse_{10}$.

5.2.2.2 Evoked Potential (EP)

For the second part of this feasibility study, the typical EP shown at the top of Figure 17 replaced the rectangle pulse and the sampling rate was increased to 100 Hz. 2000 time samples of the EEG signal were created in the same fashion as Equation 93 using the following equation:

$$
\begin{aligned}
EEG(t) = 9.5 \cdot \sin\left(2 \cdot \pi \cdot 1.2 \cdot \frac{t}{100} + 1\right) + 4.4 \cdot \sin\left(2 \cdot \pi \cdot 4.1 \cdot \frac{t}{100}\right) + \\
3.4 \cdot \sin\left(2 \cdot \pi \cdot 12.9 \cdot \frac{t}{100} + 3\right) + 2.8 \cdot \sin\left(2 \cdot \pi \cdot 16.9 \cdot \frac{t}{100} + 2\right) + \\
1.8 \cdot \sin\left(2 \cdot \pi \cdot 17.59 \cdot \frac{t}{100}\right)
\end{aligned}
\tag{98}
$$

A plot of approximately one second of the EEG generated for the second part of this feasibility study is shown in Figure 21.

The typical EP was created to last 0.2 seconds . Since the sampling rate now was 100 Hz, this equated to 20 time samples. Another random number generator was used to create 25 random start times between 1 and 2000. A start time was thrown out if it fell

within 40 time samples of another time sample. This resulted in a total of 17 EPs randomly placed throughout the EEG signal. A plot of approximately one second of the EP can be seen in Figure 22. Note in Figure 22 that the EP has two high peaks (P2 and P3) and two low peaks (N1 and N2) thus allowing something for a TDNN to pick up on for classification purposes. Five time series as shown in Figure 23 were created such that the SNR between the rectangle pulse and the EEG were different. The plots in Figure 23 are of interest to look at because the plots show that the human eye/brain can not detect the EP when the SNR $\leq$ -9.55 dB. The SNR between the EP and the EEG for each of the five time series was calculated. The effective value of the EP was approximated using the trapezoidal rule [4]. Table 7 summarizes the effective values of the EP and EEG signal in addition to the SNR between the EP and the EEG signal.

5.2.3   Methodology

A TDNN was used in both parts of this feasibility study.

5.2.3.1   Rectangle Pulse

A time lag $L = 10$ was used in a $11-25-4$ TDNN as shown in Figure 24 for rectangle pulse classification to account for the time of the rectangle pulse. The effective

Table 7. Effective Value of EP and EEG

EP Max Amplitude	$I(EP)$	EEG Max Amplitude	$I(EEG)$	SNR
200.00	6.77	20.0	8.13	-1.59 dB
63.30	5.42	20.0	8.13	-3.53 dB
20.00	4.06	20.0	8.13	-6.02 dB
6.33	2.71	20.0	8.13	-9.55 dB
2.00	1.35	20.0	8.13	-15.57 dB

Figure 21. Generated EEG at 100 Hz
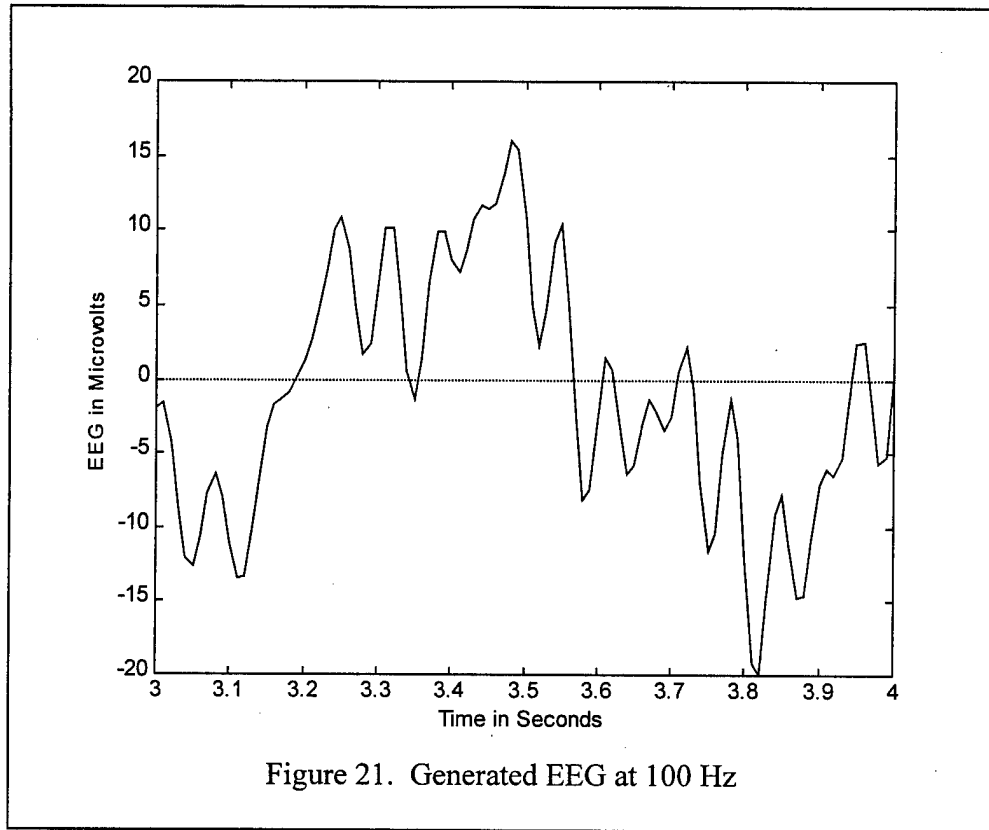
number of exemplars became $M = 1000 - 10 = 990$ due to the lags required. The training

set contained $M_{train} = 495$ exemplars, the test set contained $M_{test} = 248$ exemplars, and

the validation set contained $M_{valid} = 247$ exemplars. All inputs were standardized

following Equation 16. All weights were initialized between -0.5 and 0.5. A separate

TDNN was trained for each of the varying SNRs via instantaneous backpropagation

using a fixed learning rate $\eta = 0.3$ and no momentum.

5.2.3.2   Evoked Potential (EP)

A time lag $L = 20$ is used in a $21 - 50 - 4$ TDNN as shown in Figure 25 for EP

classification to account for the time of the EP. The effective number of exemplars

became $M = 2000 - 20 = 1980$ due to the lags required. The training set

125

Figure 22. Generated EP at 100 Hz
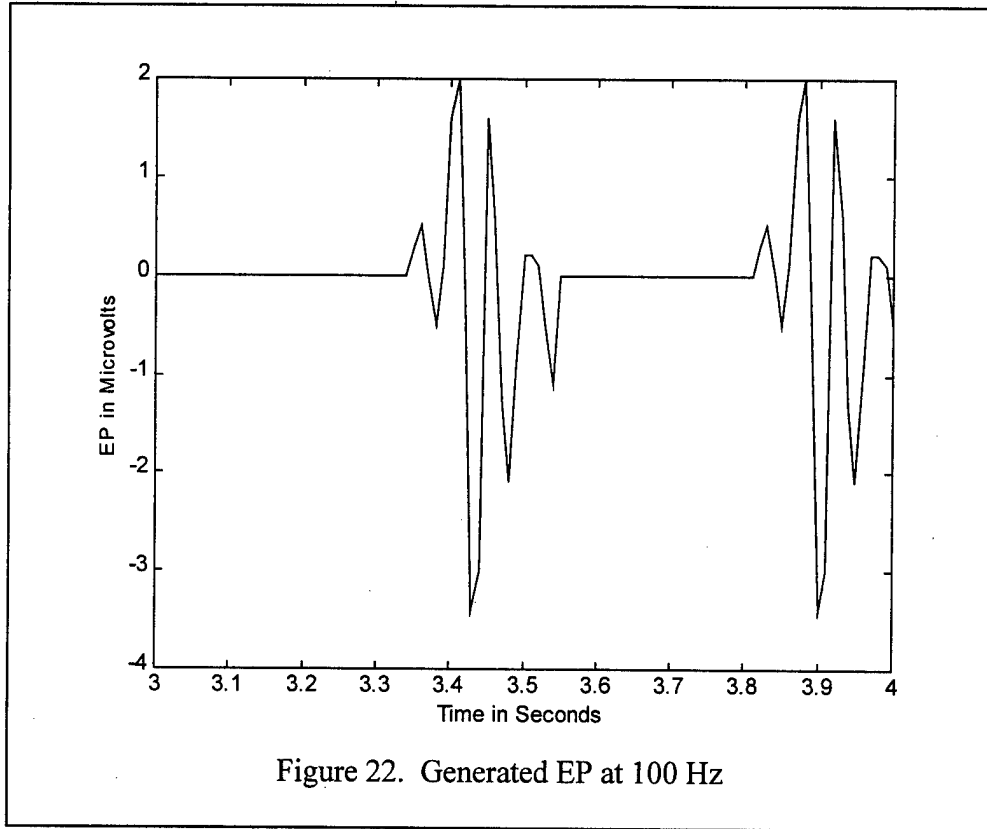
contained $M_{train} = 990$ exemplars, the test set contained $M_{test} = 495$ exemplars, and the validation set contained $M_{valid} = 495$ exemplars. All inputs were standardized following Equation 16. All weights were initialized between -0.5 and 0.5. A TDNN was trained for each of the varying SNRs via instantaneous backpropagation using a fixed learning rate $\eta = 0.3$ and no momentum.

5.2.4   Results

A total of ten TDNNs were trained. Table 8 summarizes the number of epochs required to train each TDNN in addition to the stopping rule used. Table 9 summarizes the MSE for the training, test, and validation sets for the trained TDNNs.
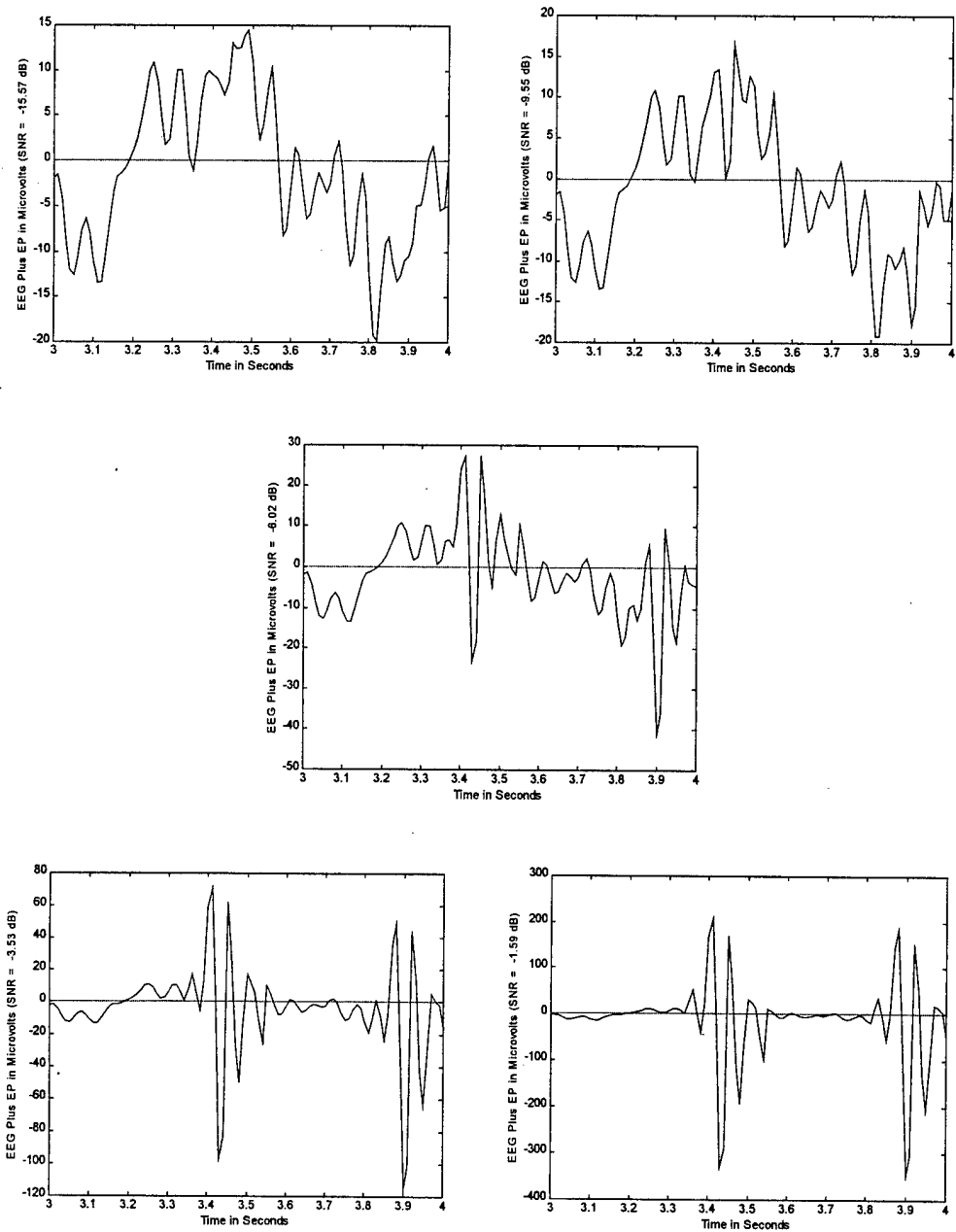
Figure 23. Generated EEG Signal with Generated EP at Varying SNRs

Figure 24. $11 - 25 - 4$ TDNN for Rectangle Pulse Classification

Figure 25. $21 - 50 - 4$ TDNN for EP Classification

128
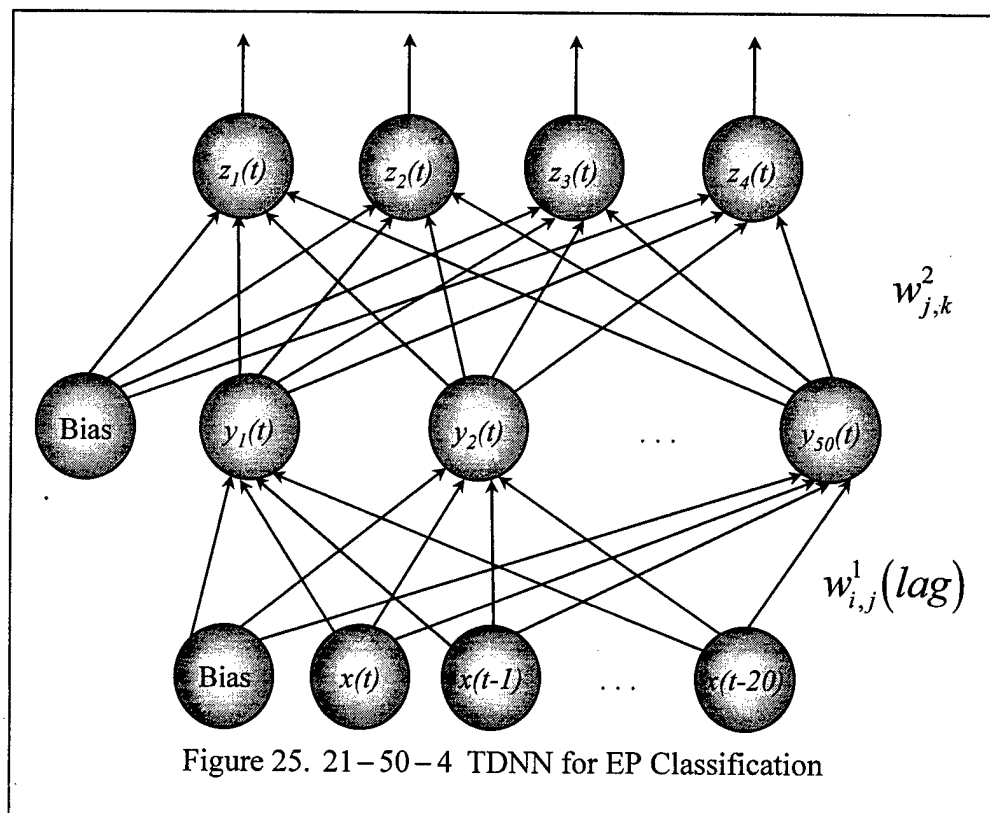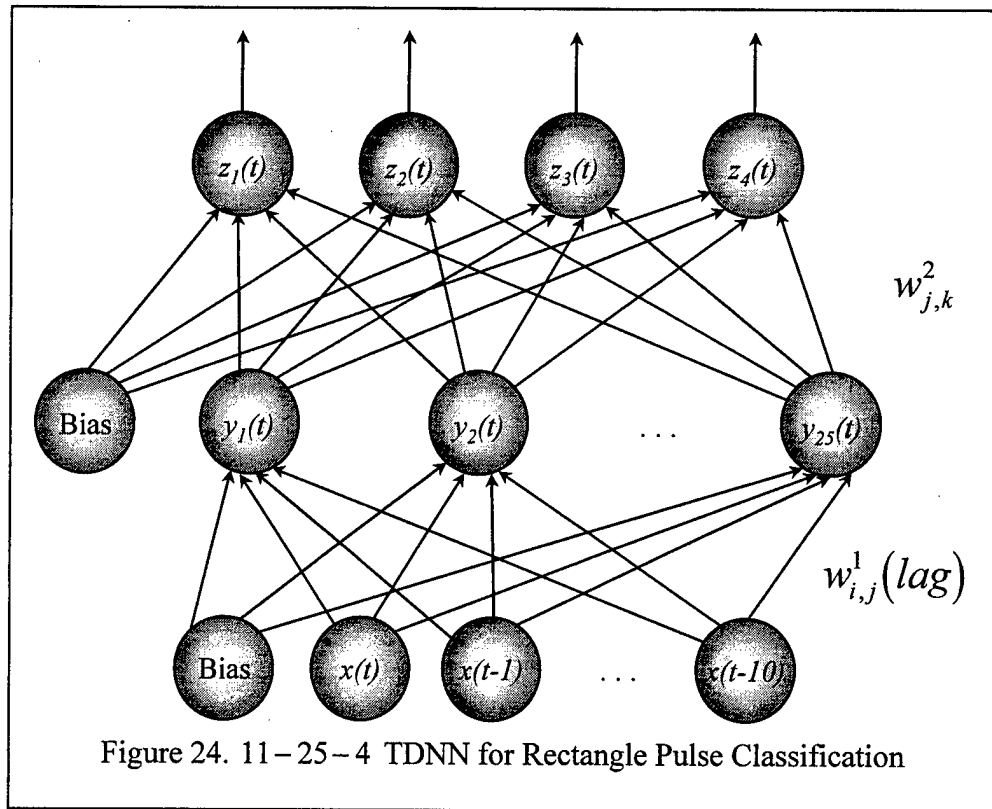
Table 8. Number of Epochs Required and Stopping Rule

Signal	SNR	Epochs	Stopping Rule
Rectangle	+27.40 dB	25	$MSE_{train} < 0.4$
Rectangle	+17.41 dB	24	$MSE_{train} < 0.4$
Rectangle	+7.40 dB	99	$MSE_{train} < 0.4$
Rectangle	-2.59 dB	779	$MSE_{train} < 0.4$
Rectangle	-12.60 dB	168	Minimum MSE_{test}
EP	-1.59 dB	62	$MSE_{train} < 0.4$
EP	-3.53 dB	240	$MSE_{train} < 0.4$
EP	-6.02 dB	723	$MSE_{train} < 0.4$
EP	-9.55 dB	1352	No significant change in MSE_{train} or MSE_{test} in over 500 epochs
EP	-15.57 dB	1308	Minimum MSE_{test}

Figure 26 provides plots of the information provided in Table 9. Table 10 summarizes the *CA* for the training, test, and validation sets for the trained TDNNs. Figure 27 provides plots of the information provided in Table 10.

5.2.4.1 Rectangle Pulse

The TDNNs for rectangle pulse classification performed adequately when the SNR was +27.40 dB, +17.41 dB, or 7.40 dB. When the SNR was -2.59 dB or -12.60 dB,

Figure 26. *MSE* for Varying SNRs
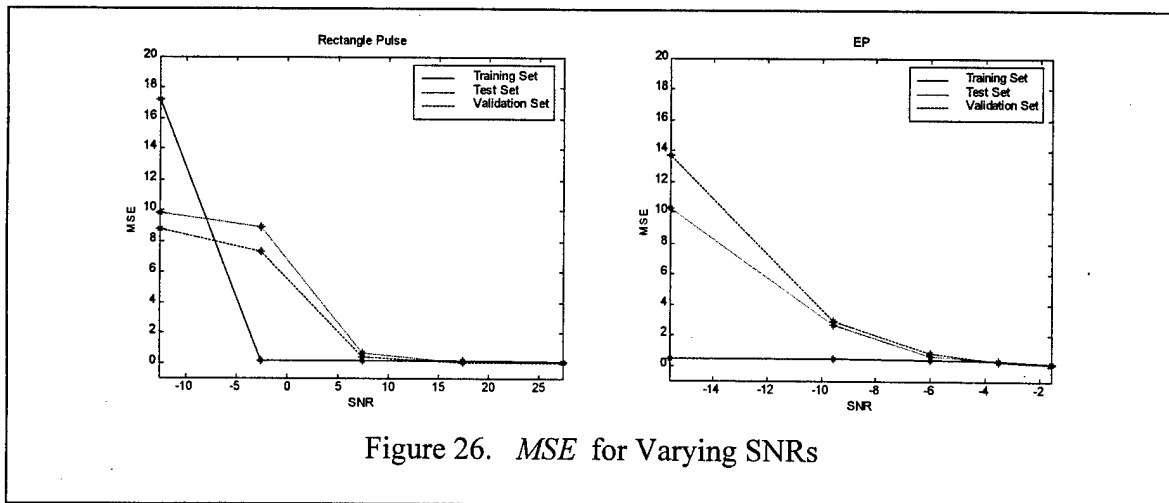
Table 9. *MSE*

Signal	SNR	Training	Test	Validation
Rectangle	+27.40 dB	0.13	0.06	0.04
Rectangle	+17.41 dB	0.15	0.07	0.06
Rectangle	+7.40 dB	0.18	0.67	0.40
Rectangle	-2.59 dB	0.19	8.92	7.34
Rectangle	-12.60 dB	17.24	9.84	8.77
EP	-1.59 dB	0.17	0.12	0.12
EP	-3.53 dB	0.34	0.33	0.31
EP	-6.02 dB	0.40	0.66	0.86
EP	-9.55 dB	0.51	2.68	2.97
EP	-15.57 dB	0.45	10.26	13.74

the TDNN for rectangle pulse classification did not perform adequately. At -2.59 dB, the TDNN did not performed adequately on the validation set and is evidenced by only 18.75% of the Class 2 exemplars being correctly classified, 22.22% of the Class 3 exemplars being correctly classified, and 16.67% of the Class 4 exemplars being correctly classified. In the majority of misclassifications at the -2.59 dB level, the exemplar was misclassified as belonging to Class 1.

The TDNN for rectangle pulse classification at -12.60 dB did not perform adequately on its validation set, either. In fact, 0.00% of the Class 2, Class 3, and Class 4

Table 10. *CA*

Signal	SNR	Training	Test	Validation
Rectangle	+27.40 dB	100.00%	100.00%	100.00%
Rectangle	+17.41 dB	100.00%	100.00%	100.00%
Rectangle	+7.40 dB	99.80%	95.56%	97.98%
Rectangle	-2.59 dB	98.99%	83.06%	83.40%
Rectangle	-12.60 dB	89.09%	86.29%	86.64%
EP	-1.59 dB	100.00%	99.80%	100.00%
EP	-3.53 dB	97.98%	96.77%	96.97%
EP	-6.02 dB	96.97%	95.76%	94.75%
EP	-9.55 dB	95.96%	92.93%	91.31%
EP	-15.57 dB	98.08%	87.68%	84.85%
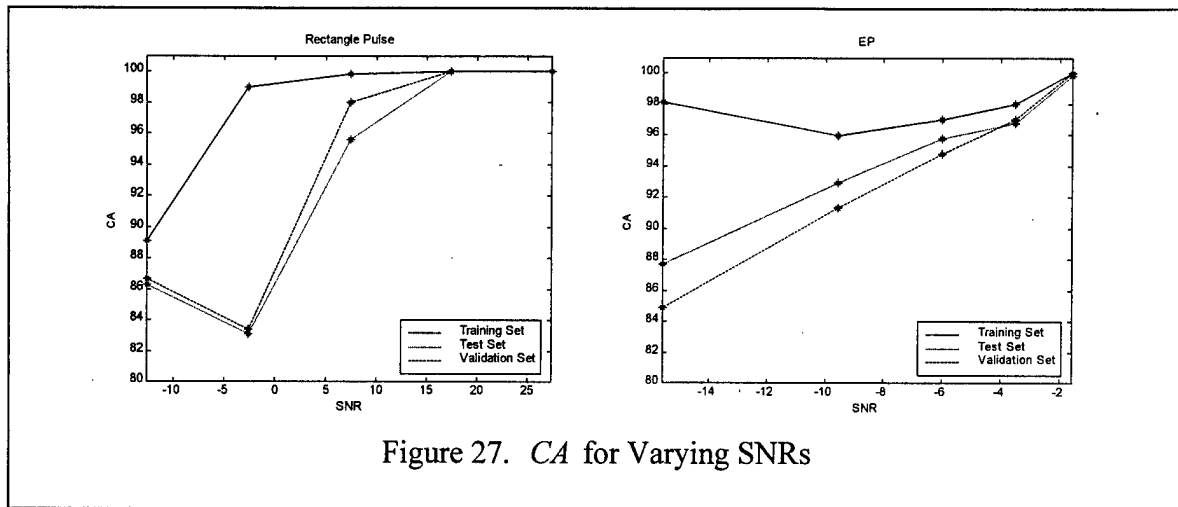
Figure 27. *CA* for Varying SNRs

validation exemplars were correctly classified. All but two validation exemplars were classified as belonging to Class 1.

5.2.4.2 Evoked Potential (EP)

The TDNNs for EP classification performed adequately when the SNR was -1.59 dB, -3.53 dB, -6.02 dB, and -9.55 dB. The TDNN did surprisingly well when the SNR was -9.55 dB. The TDNN for EP classification at -15.57 dB did not perform adequately but it performed better than the TDNN for rectangle pulse classification at -12.82 dB. At -15.57 dB, the TDNN for EP classification correctly classified 96.89% of the validation Class 1 exemplars, 14.29% of the Class 2 exemplars, 26.67% of the Class 3 exemplars, and 15.79% of the Class 4 exemplars.

5.2.5 Conclusions

Since the actual SNR between an EP and EEG is -20 dB, it is clear from this feasibility study that the modeling of pilot workload in addition to air traffic controller workload should not utilize single event EPs. Potential features for classifying mental

workload are the amplitudes and latencies of the EP components (i.e. N1, P2, N2, and P3). Wilson et al. showed that the averaged amplitude of the N1 and P2 components of F-4 pilots performing the oddball paradigm like that seen in Figure 17 were significantly smaller in flight [175]. Wilson et al. also showed that the averaged amplitude of the P2 component of F-4 pilots performing the oddball paradigm like that seen in Figure 17 was significantly smaller if the pilot was flying the airplane as opposed to the weapon systems operator (WSO) flying the airplane [175]. In addition, Wilson et al. concluded that the averaged amplitude of the P3 component of F-4 pilots performing the oddball paradigm like that seen in Figure 17 was significantly larger for the particular tone that the pilot was instructed to count [175]. The results obtained by Wilson et al. were averaged over seven F-4 pilots and over 100 trials [175]. It is necessary to first identify that an EP is present before determining the amplitude of its components.

As with anything, a lot more work could be done on the pattern recognition of EPs in a EEG signal. Future research in this area, though not conducted in this dissertation, may improve upon several things. This feasibility study did not consider optimization of the number of time lags L. Applying Taken's Theorem as in Equation 57 can provide an upper and lower bound to L using the fractal dimension of the EEG signal, the EP, or the EEG signal with the EP embedded. Another idea may be to utilize saliency screening methods as described in Section 3.5 to help select the optimal L.

This feasibility study simply used the raw amplitude of the time series. There may be other features that can provide valuable information to a TDNN for classifying an EP. For example, an average of the time samples over a fixed window may be utilized in an attempt to smooth the "noise" of the EEG. The standard deviation of a fixed number

of time samples may also provide a measure of the fluctuations in the time series so that a high standard deviation may flag the presence of an EP starting or an EP ending and a lo standard deviation would indicate the presence of no EP.

An EEG signal with a rectangle pulse randomly placed throughout can be classified when the SNR is +27.40 dB, +17.41 dB, or 7.40 dB. An EEG signal with an EP randomly placed throughout can be classified when the SNR is -1.59 dB, -3.53 dB, -6.02 dB, and -9.55 dB. The actual SNR between a typical EP and EEG is -20 dB. In conclusion, a TDNN will more than likely not be able to classify single event EPs in real EEG data.

5.3 Feasibility of Using Elman Recurrent Neural Networks (RNN) to Classify Mental Workload Using Ongoing Electroencephalography (EEG)

5.3.1 Introduction

The purpose of this feasibility study was to investigate the use of Elman RNNs for classifying mental activity using EEG in the presence of noise. If an Elman RNN is ever to classify pilot workload using EEG collected during flight, than an Elman RNN classifier must be robust to the effects of noise. There are many sources of potential noise in a cockpit including vibration, movement, talking on the radios, and G forces. For this feasibility study, EEG was collected from a test subject performing three types of mental activity. An Elman RNN was first trained using 10 features derived from the α-band to classify the type of mental activity being performed. Ten test sets with varying levels of noise were used to evaluate the Elman RNN's robustness to noise. The

MOE was CA_{test}. Next, an Elman RNN was trained using 90 features derived from the nine frequency bands listed in Table 4. Again, 10 test sets with varying levels of noise were used to evaluate the Elman RNN's robustness to noise.

5.3.2 Data

The test subject used in this feasibility study is a 50-year old male who is in excellent health and takes no medications. EEG was collected from the test subject at the Flight Psychophysiological Laboratory, Wright-Patterson AFB, OH. The WAM recorded EEG from six electrodes according to Figure 15 and two reference electrodes following the International 10-20 standard at a sampling rate of 128 Hz. The first mental task for the test subject was to read an article from *Science* magazine. He was told that there would be a quiz after data collection so that he concentrated on reading the material (he really was never given a quiz though). The reading task was performed for three minutes. The test subject's next task was to sit quietly but with his eyes open for three minutes. The third and final task was to sit quietly but with his eyes closed for three minutes. It was soon discovered, unfortunately, that electrode O1 did not pick up a signal and was thus removed from the data set.

The WAM preprocessed the EEG signals as described in Section 4.4. The preprocessed data from the WAM was then further processed using *MATLAB* code as described in Section 4.4 to calculate the log of the power and the variance of the power over a moving 10-second window with 50% overlap for each frequency band for each electrode. The frequency bands as listed in Table 4 were used. There were a total of 34 exemplars for each of the three classes of mental activity. Overall, there were 102

exemplars. Figure 28 shows the log power of the α-band for each of the five electrodes broken into the three classes of mental activity. A unit on the x-axis of each subfigure represents a 10-second moving window. In addition, the dotted line in each of Figure 28's subfigures shows the mean log power of the α-band for the corresponding mental activity class. Figure 29 shows the variance of the log power of the α-band for each of the five electrodes broken into the three classes of mental activity. As in Figure 28, a unit on the x-axis of each subfigure represents a 10-second moving window. In addition, the dotted line in each of Figure 29's subfigures show the mean variance of the log power of the α-band for the corresponding mental activity class. The log power of the α-band as shown in Figure 28 and that of the variance of the log power of the α-band as shown in Figure 29 is representative of the other EEG frequency bands used in this feasibility study.

5.3.3 Methodology

5.3.3.1 Ten Input Features

The first Elman RNN trained as shown in Figure 30 had a $10 + 20/20/3$ architecture. Only the α-band features were used as inputs to the first Elman RNN trained. It is expected that the log power of the α-band will increase as the test subject transitions from "reading" to "eyes open" to "eyes closed." The log power of the α-band for all five electrodes exhibits this behavior in Figure 28. There were a total of 10 input features to the Elman RNN representing the log power of the α-band and the variance of the log power of the α-band from five electrodes. Each input feature was normalized between 0.0 and 1.0 following Equation 19.

Electrode Fz

Electrode Cz

Electrode Pz

Electrode T5
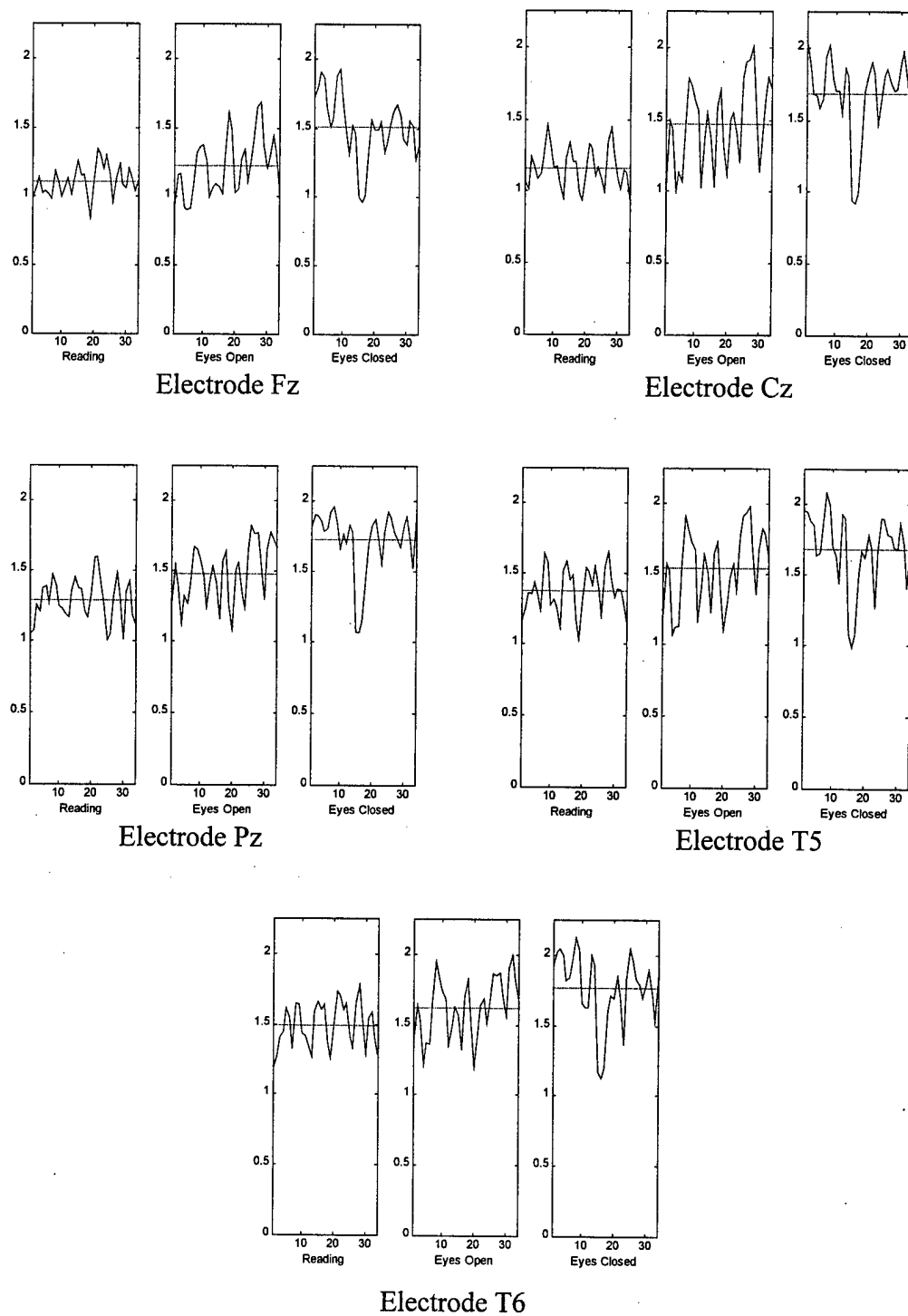
Electrode T6

Figure 28. Log Power of α-Band
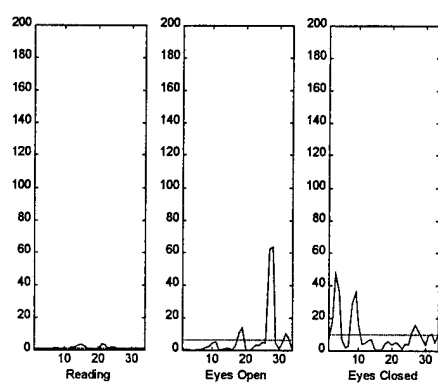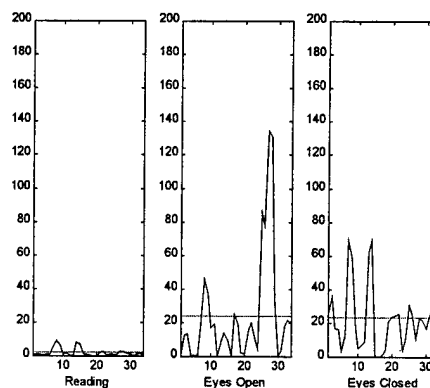
Electrode Fz   Electrode Cz
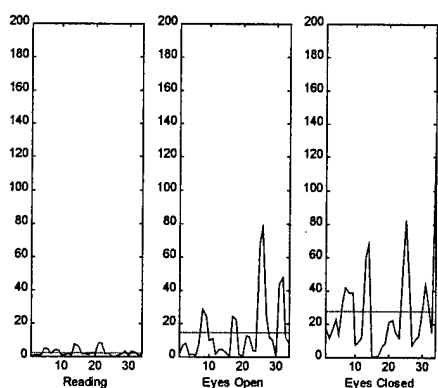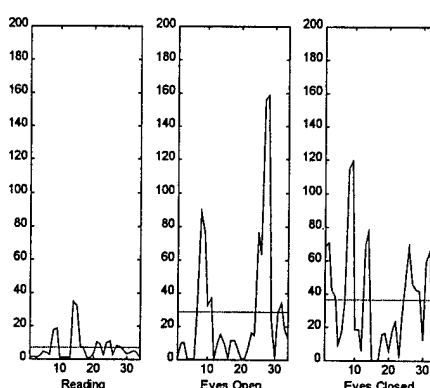
Electrode Pz   Electrode T5
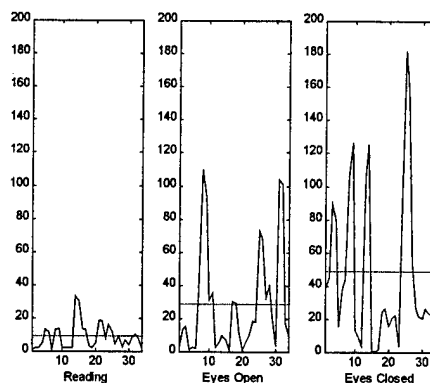
Electrode T6

Figure 29. Variance of Log Power of α-Band

Figure 30. First Elman RNN Architecture Attempted
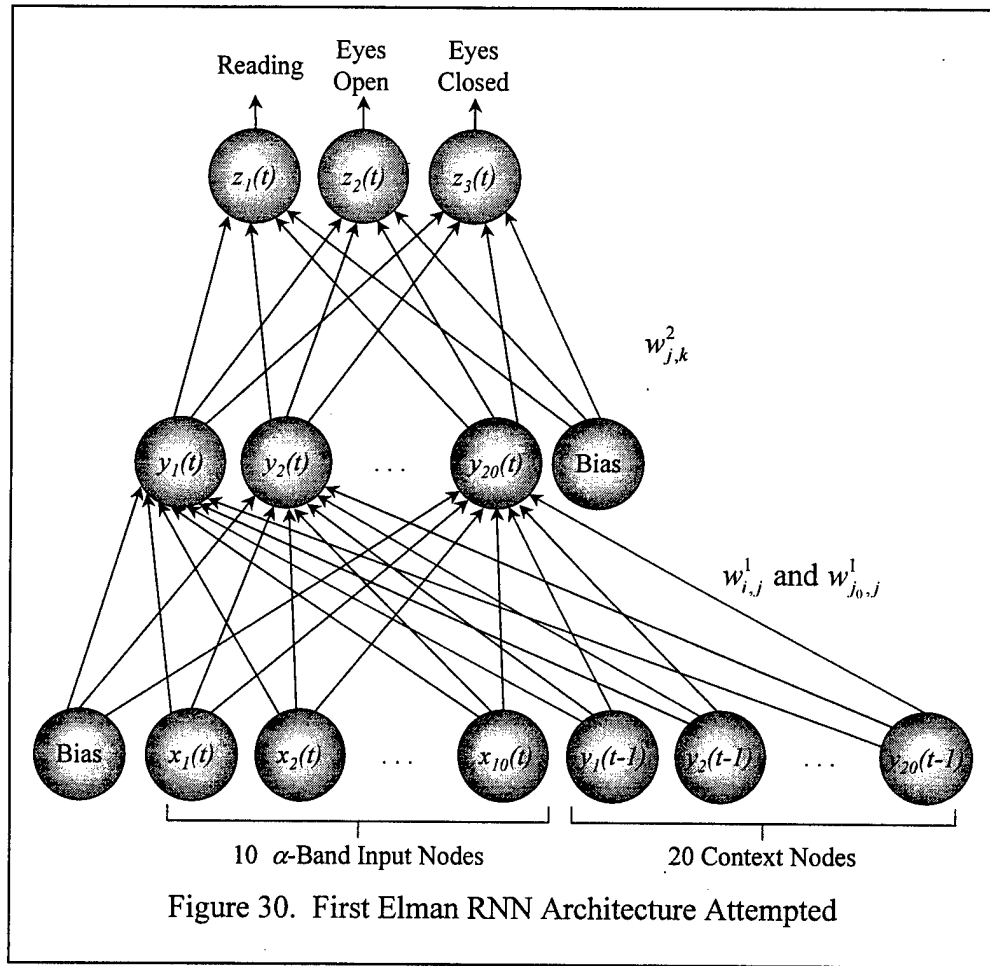
Twenty context nodes were used. There were three output classes: reading, eyes open, and eyes closed. The hidden/context nodes were activated by the sigmoid nonlinear transfer function. The output nodes were activated by a linear transfer function with slope = 1. The Elman RNN was trained using backpropagation with momentum and an adaptive learning rate following Equation 50. The initial learning rate η was set to 0.001. Training was stopped after 35,000 epochs.

Ten test sets were created with different levels of added noise to test the Elman RNN trained using 10 features from the α-band. For each of the 10 tests sets, noise following a Uniform random distribution was added to each normalized input feature in

the training set. For the first test set, noise following a Uniform random distribution between 0.00 and 0.05 denoted as U(0.00,0.05) was added to each input feature. For the second test set, noise following a U(0.00,0.10) distribution was added to each input feature. For the third test set, noise following a U(0.00,0.15) distribution was added to each input feature. And so on. Table 11 describes the maximum value of the Uniformly distributed noise added to each input feature for each test set.

5.3.3.2 Ninety Input Features

The next Elman RNN trained as shown in Figure 31 had a $90 + 180 / 180 / 3$ architecture. It used all 90 features available from the collected EEG data. The 90 features represent the log power and the variance of the log power from nine frequency bands collected from five electrodes. All input features were normalized between 0.0 and 1.0 following Equation 19. 180 context nodes were used. The were three output classes: reading, eyes open, and eyes closed. The hidden/context nodes were activated by the sigmoid nonlinear transfer function. The output nodes were activated by a linear transfer function with slope = 1. The Elman RNN was trained using backpropagation with momentum and an adaptive learning rate following Equation 50. The initial learning rate η was set to 0.001. Training was stopped after 10,500 epochs.

Ten test sets were created with different levels of added noise to test the Elman RNN trained using 90 features in the same fashion as described in Section 5.3.3.1.

Table 11. Maximum Value of Uniform Distribution for Testing

Test Set	1	2	3	4	5	6	7	8	9	10
Max(Uniform)	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50

Figure 31. Second ERNN Architecture Attempted
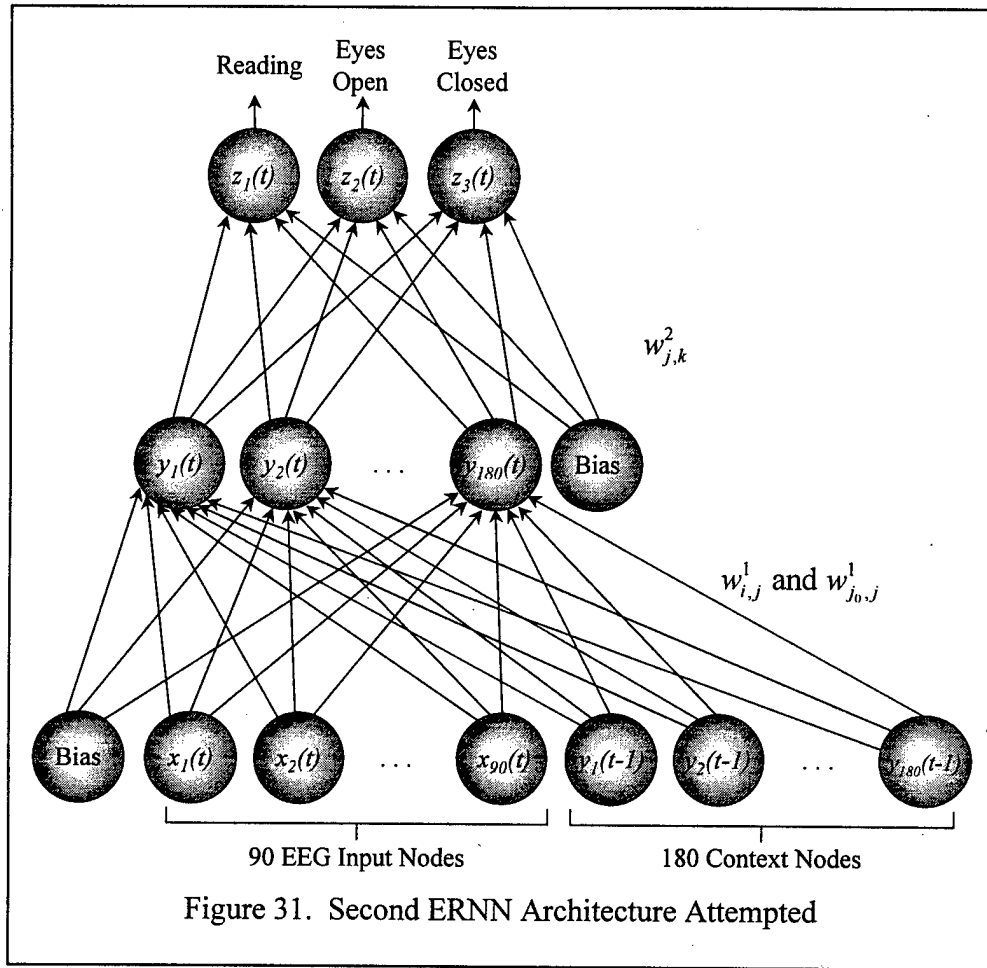
5.3.4 Results

5.3.4.1 Ten Input Features

Training was stopped after 35,000 epochs which took about five hours on a Pentium-90 personal computer (PC). The SSE_{train} was 1.70 and the CA_{train} was 100%. The solid curve in Figure 32 is the CA_{test} using 10 features for each level of noise added. About 95% of the misclassifications from the test sets were the results of the Elman RNN misclassifying the mental activity as "reading" instead of "eyes open." This implies that the log power and the variance of the log power of the α-band from five electrodes may not be enough to separate "reading" from "eyes open" for this 50-year old male test

subject. The other 5% or so of the misclassifications occurred at the transitions from "reading" to "eyes open" or from "eyes open" to "eyes closed."

5.3.4.2 Ninety Input Features

Training was stopped after 10,500 epochs which took about five hours on a Pentium-133 PC. The SSE_{train} was 0.90 and the CA_{train} was 100%. The dashed curve in Figure 32 is the CA_{test} using 90 input features for each level of noise added. There does not appear to be any trend associated with the misclassifications using 90 input features. By comparing the two curves in Figure 32, it appears that an Elman RNN that includes all 90 features is more robust to noise.

Figure 32. CA_{test} for Differing Levels of Noise Added
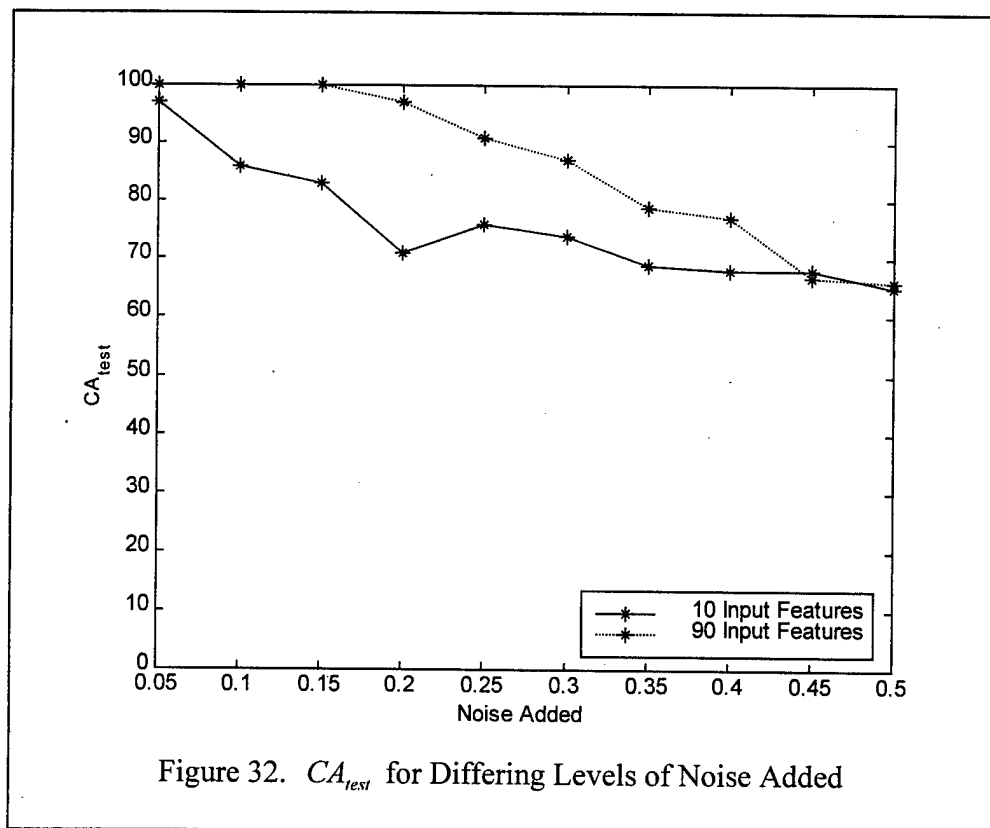
5.3.5 Conclusions

This feasibility study shows that an Elman RNN can adequately classify among three types of mental activity even in the presence of added noise. In both Elman RNNs trained, the CA_{train} with no added noise was 100%. With only 10 input features derived from the α-band, the CA_{test} remains greater than 80% so long as the noise added is no larger than 0.15. With all 90 input features, the CA_{test} remains greater than 80% so long as the noise added is no larger than 0.30. The Elman RNN trained with 90 features appears to be more robust to the effects of added noise. The Elman RNN shows promise for classifying pilot workload in addition to air traffic controller workload.

6 Signal-to-Noise Ratio (SNR) Saliency Measure as Applied to Classifying the Workload of Pilots in Addition to Air Traffic Controllers via Feedforward Multilayer Perceptron (MLP) Artificial Neural Networks (ANN)

6.1 Introduction

The SNR saliency measure is a new saliency measure. The SNR saliency measure determines the saliency, or relative importance, of a feature by comparing it to an injected noise feature. Bauer proposed the SNR saliency measure [6] and Sumrell was the first to experiment with the SNR saliency measure using a noisy version of the XOR classification problem as shown in Figure 2 and Fisher's iris classification problem [147].

This chapter summarizes the application of both partial derivative-based saliency measures in Equations 74 and 76, the weight-based saliency measures in Equations 77 through 80, and the SNR saliency measure to classify pilot workload in addition to air traffic controller workload via feedforward MLP ANNs as published in [46] and submitted for publication in [50]. This dissertation research produced the first non-trivial, real-world applications of the SNR saliency measure.

This research summarized in this chapter had two objectives. The primary objective was to develop a methodology to identify salient features to classify the workload of pilots in addition to air traffic controllers. The second objective was to compare the results of the SNR saliency measure to that of partial derivative-based and weight-based saliency measures.

6.2 Signal-to-Noise Ratio (SNR) Saliency Measure

The SNR saliency measure directly compares the saliency of a feature to that of

an injected noise feature. The theoretical basis for a comparison to noise is similar to previous research performed by Belue [11, 12] and Steppe [136, 137, 138, 139] as described in Sections 3.5.2 and 3.5.3. The SNR saliency measure is computed using the first layer weights of a trained ANN as the following:

$$SNR_i = 10 \cdot \log\left(\frac{\sum_{j=1}^{J} \left(w_{i,j}^1\right)^2}{\sum_{j=1}^{J} \left(w_{N,j}^1\right)^2} \right) \tag{99}$$

where SNR_i is the value of the SNR saliency measure for feature $i = 1, 2, \ldots, I$ and $w_{N,j}^1$ is the first layer weight from the injected noise node N to hidden node j. The injected noise feature is created such that its distribution follows that of a U(0.00,1.00) random variable. All feature inputs are normalized (or standardized) so that the features are "unitless" thus preventing the input features with larger value from dominating. The scaled logarithm transformation of the ratio converts the saliency measure to a decibel scale. The effect of the scaled logarithm transformation is shown in Figure 33. It is very interesting to compare Figure 33 with the penalty function employed by Setiono-Liu in Figure 13. The effect of the scaled logarithm transformation appears to be similar to that of the penalty function employed by Setiono-Liu. A small ratio will produce a near-zero or negative SNR saliency measure. On the other hand, a large ratio will produce a large SNR saliency measure that increases as a logarithmic function of the ratio. Note that negative ratio values are not possible due to squared terms but are shown in Figure 33 to illustrate similarity to Figure 13.

The SNR saliency measure is a weight-based saliency measure since it relies on the sum of squared first layer weights. The theoretical concept behind the SNR saliency

144

Figure 33. Effect of $10 \cdot \log$ On Weight Ratio
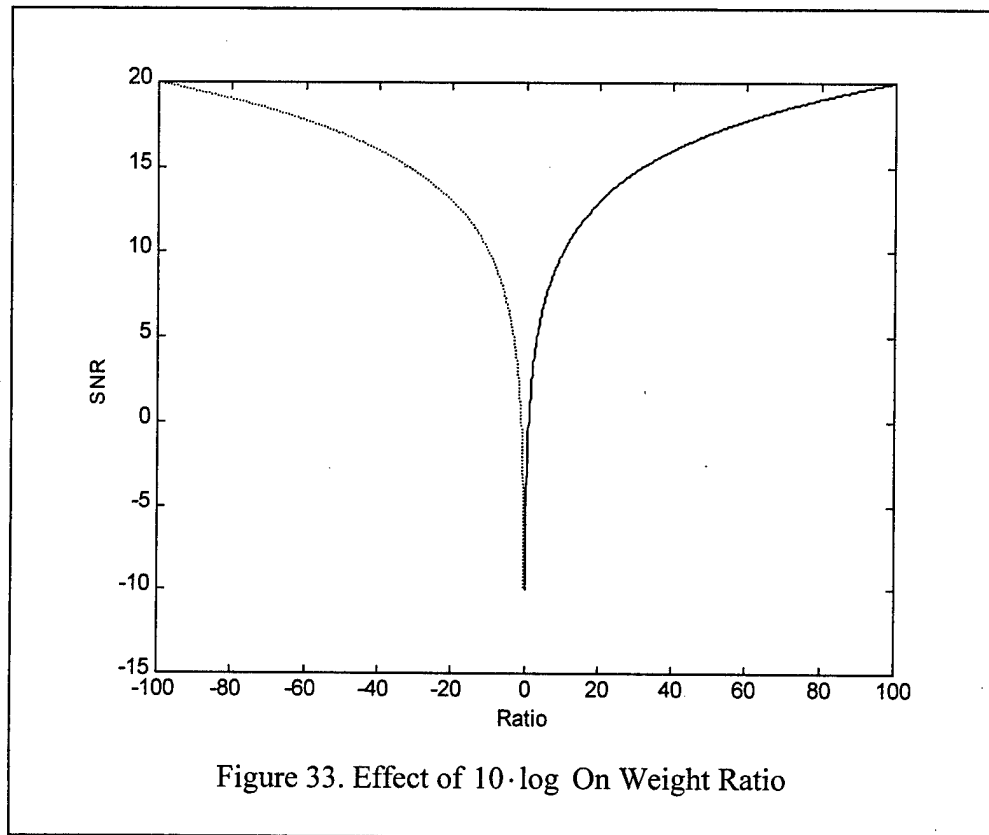
measure is similar to that of the weight-based saliency measures as described in Section 3.4.3. If a given feature is not relevant to an ANN's output, the updates of the first layer weights emanating from that feature's input node should be random and simply fluctuate around zero [152]. If, on the other hand, a given feature is relevant to an ANN's output, then the updates of the first layer weights emanating from that feature's input node should be moved in the weight space in a constant direction until the error is minimized [152]. Thus, the SNR saliency measure should be significantly larger than 0.0 for salient features and very close to 0.0 or less than 0.0 for nonsalient features.

The SNR saliency measure is similar to weight-based saliency measures as described in Section 3.4.3 in that it relies on the sum of squared first layer weights. The SNR saliency measure, however, is different from weight-based saliency measures in

145

addition to partial derivative-based saliency measures described in Section 3.4.2 because the SNR saliency measure directly compares the saliency of a feature to that of an injected noise feature.

The SNR saliency measure is appealing because, like the partial derivative-based and previously discussed weight-based saliency measures, the SNR saliency measure can be used to rank order the features from most relevant to least relevant. The greatest potential of the SNR saliency measure results from its comparison of the saliency of each feature to a baseline noise feature.

6.3 Classifying Pilot Workload

6.3.1 Introduction

Before this dissertation research, a set of salient features had never been identified for classifying pilot workload. In previous studies, psychophysiological features were selected by maximization of the *CA* by trial and error [175]. For classifying pilot workload, no one psychophysiological feature has been demonstrated as sufficient and no one psychophysiological feature has been demonstrated as superior. Individual differences in psychophysiological response and in particular mental response should be recognized. As a result, an extremely large number of psychophysiological features were used as inputs to feedforward MLP ANNs in previous research efforts.

6.3.2 Data

The pilot workload data set consisted of processed data from a flight simulation. The test subject flew a simulated aircraft landing scenario. The scenario started off with

146

the test subject flying his descent in the clouds. While in the clouds, the pilot's workload was classified as low. The test subject then broke through the clouds on his descent to the airfield. As soon as the pilot broke through the clouds, his workload was classified as high. The scenario ended at touchdown.

Throughout the scenario, the WAM collected EEG at the six scalp locations shown in Figure 16. The EEG data were preprocessed as described in Section 4.4 and then grouped into nine different bands as summarized in Table 4. For each band at each electrode, two general categories of features were calculated over a 10-second moving window resulting in a total of 108 features (9 bands x 6 electrodes x 2 types). For every electrode and frequency band, the two general categories of features computed were:

- Log power averaged over a 10-second moving window with 50% overlap.
- Variance of the power over a 10-second moving window with 50% overlap.

Four peripheral psychophysiological features derived from EOG, ECG, and respiration gauges were also developed for inclusion in the data set. One feature, number of eye blinks in a 10-second moving window with 50% overlap, was derived from EOG. One respiratory feature, interbreath interval averaged over a 10-second moving window with 50% overlap, was derived from the respiration gauges. Two cardiopulmonary features were derived from the ECG:

- Interbeat interval averaged over a 10-second moving window with 50% overlap.
- Slope of the heart rate over a 10-second moving window with 50% overlap.

The input feature set contained a total of $I = 112$ psychophysiological features (108 EEG features and 4 peripheral psychophysiological features). Unfortunately, the input feature set contained only $M = 32$ exemplars: 22 high workload exemplars and 10 low workload exemplars. Foley's Rule as given in Equation 59 and Cover's Theorem as

given in Equation 60 applied to this two-class (low workload, high workload) problem. Since the data set contains 112 features and only 32 exemplars, it was clear that both Foley's Rule and Cover's Theorem are violated. Two solutions existed:

1. Increase the number of exemplars.
2. Decrease the number of features.

It was desired to decrease the number of features by selecting the most salient features.

6.3.3 Methodology

As a first attempt to screen the input feature set, an analysis of the correlation between each feature and pilot workload was conducted. The sample correlation between each feature x_i for $i = 1,2,...,I$ and the desired workload d was computed as:

$$\rho_{i,d} = \frac{C(i,d)}{S_i \cdot S_d} \tag{100}$$

where $\rho_{i,d}$ is the sample correlation between feature $i = 1,2,...,I$ and the desired workload d where $d = 0$ for low workload and $d = 1$ for high workload, $C(i,d)$ is the sample covariance between feature $i = 1,2,...,I$ and d, S_i is the sample standard deviation of feature $i = 1,2,...,I$, S_d is the sample standard deviation of d, and $I = 112$. The sample covariance $C(i,d)$ was computed as:

$$C(i,d) = \frac{1}{M-1} \cdot \sum_{m=1}^{M} (x_{i,m} - \overline{x}_i) \cdot (d_m - \overline{d}) \tag{101}$$

where $M = 32$ exemplars, $\overline{x}_i$ is the mean of feature $i = 1,2,...,I$, and $\overline{d}$ is the mean of the d so that:

$$\bar{x}_i = \frac{\sum_{m=1}^{M} x_{i,m}}{M} \qquad \bar{d} = \frac{\sum_{m=1}^{M} d_m}{M} \qquad (102)$$

The sample standard deviations for feature x_i for $i = 1,2,\ldots,I$ and for d were computed as:

$$S_i = \sqrt{\frac{\sum_{m=1}^{M}\left(x_{i,m} - \bar{x}_i\right)^2}{M-1}} \qquad S_d = \sqrt{\frac{\sum_{m=1}^{M}\left(d_m - \bar{d}\right)^2}{M-1}} \qquad (103)$$

Those features with correlation coefficients $\rho_{i,d} < 0.75$ were removed and no longer considered as possible salient features.

$M_{train} = 22$ exemplars were randomly selected as training exemplars and the remaining $M_{test} = 10$ exemplars were used as test exemplars. A "noise" feature with a U(0.0,1.0) distribution was added to the data set for use as a baseline and for use in the SNR saliency measure. Each of the input features were normalized between 0.0 and 1.0 following Equation 19. The weights were initialized between -0.001 and 0.001 in order to equalize the signal strength for all input features. All hidden and output nodes are activated by sigmoid nonlinear transfer functions. A $19-19-2$ feedforward MLP ANN was trained via instantaneous backpropagation with a fixed learning rate $\eta = 0.3$ and no momentum until all of the following stabilized:

- MSE_{train}
- MSE_{test}
- **W**
- $SNR_i \ \forall \ i = 1,2,\ldots,I$.

6.3.4 Results

Eighteen of the original 112 features had a sample correlation $\rho_{x_i,d} \geq 0.75$. Seventeen of the selected features were EEG features. Of those 17 EEG features selected in the preliminary screening, ten were average power features and seven were variance features. The only peripheral psychophysiological feature selected in the preliminary screening was number of eye blinks.

The feedforward MLP ANN was trained in 1200 epochs using 18 input features plus the injected noise feature. The CA_{test} was 90.0%. Table 12 lists all of the saliency measures computed based upon the following:

- Correlation ($\rho_{i,d}$)
- Partial derivative-based saliency measure (Λ_i)
- Partial derivative-based saliency measure with pseudo-sampling ($\hat{\Lambda}_i$)
- Weight-based saliency measure (τ_i)
- Euclidean norm of weight-based saliency measure ($\tau_i^{\nu 1}$)
- Taxi-Cab norm of weight-based saliency measure ($\tau_i^{\nu 2}$)
- Infinity norm of weight-based saliency measure ($\tau_i^{\nu 3}$)
- SNR saliency measure (SNR_i)

If we adhere to Foley's Rule given that there are only $M = 32$ exemplars, at most $I = 5$ features (3 Foley's Factor x 5 features x 2 classes) could be used to train the feedforward MLP ANN for this two-class pilot workload problem. If $I = 5$ features were selected, then $M_{train} = 30$ training exemplars must be randomly selected leaving only $M_{test} = 2$ test exemplars. This would not adequately test the feedforward MLP ANN. Thus, only $I = 4$ features were selected to train a feedforward MLP ANN using $M_{train} = 24$ training exemplars (3 Foley's Factor x 4 features x 2 classes) and $M_{test} = 8$ test exemplars.

150

Table 12. Calculated Feature Saliency Measures

i	$\rho_{i,d}$	Λ_i	$\hat{\Lambda}_i$	τ_i	τ_i^{v1}	τ_i^{v2}	τ_i^{v3}	SNR_i
Noise	0.0187	0.0001	0.0001	0.01	0.10	0.46	0.02	0.0000
$\mathrm{Var}(\alpha_1)$ at FZ	0.7535	0.0005	0.0008	0.78	0.88	3.94	0.20	42.7731
$\mathrm{Var}(\alpha_2)$ at FZ	0.7570	0.0004	0.0006	0.58	0.76	3.41	0.17	39.8555
$\mathrm{Var}(\alpha)$ at FZ	0.7684	0.0004	0.0007	0.63	0.80	3.56	0.18	40.7045
$\mathrm{Var}(\alpha)$ at C4	0.8039	0.0005	0.0008	0.88	0.94	4.20	0.22	44.0242
$\mathrm{Var}(\alpha)$ at P3	0.7514	0.0005	0.0008	0.80	0.89	3.99	0.21	43.0018
$\mathrm{Var}(\beta_1)$ at FZ	0.7801	0.0005	0.0007	0.77	0.88	3.93	0.20	42.6814
$\mathrm{Var}(\beta_1)$ at C4	0.7919	0.0004	0.0007	0.66	0.81	3.63	0.19	41.1027
$\mathrm{Log}(\alpha_2)$ at FZ	0.8989	0.0006	0.0010	1.29	1.13	5.07	0.26	47.7967
$\mathrm{Log}(\alpha_2)$ at C4	0.8620	0.0006	0.0009	1.19	1.09	4.87	0.25	46.9922
$\mathrm{Log}(\alpha)$ at C4	0.8164	0.0006	0.0009	1.14	1.07	4.78	0.24	46.6283
$\mathrm{Log}(\beta_1)$ at C4	0.8426	0.0006	0.0009	1.02	1.01	4.52	0.23	45.4791
$\mathrm{Log}(\Delta)$ at FP1	0.8607	0.0006	0.0010	1.35	1.16	5.19	0.27	48.2794
$\mathrm{Log}(\Delta)$ at FZ	0.8731	0.0007	0.0010	1.45	1.20	5.38	0.27	48.9860
$\mathrm{Log}(\Delta)$ at C4	0.8827	0.0006	0.0010	1.40	1.18	5.29	0.27	48.6425
$\mathrm{Log}(\Delta)$ at P3	0.9702	0.0007	0.0011	1.71	1.31	5.85	0.30	50.6575
$\mathrm{Log}(\Delta)$ at T6	0.8577	0.0006	0.0010	1.28	1.13	5.06	0.26	47.7640
$\mathrm{Log}(\theta)$ at FZ	0.7921	0.0006	0.0009	1.10	1.05	4.69	0.24	46.2252
Number Eye Blinks	0.8965	0.0007	0.0010	1.51	1.23	5.50	0.28	49.4052
Constant	NA	0.0002	0.0003	0.10	0.31	1.41	0.07	22.1355

Table 13 summarizes the top four rankings for the partial derivative-based, weight-based, and SNR saliency measures and Table 14 describes the four most salient features. From examination of Table 13, it appears that the saliency measures provided consistent rankings. All of the saliency measures ranked feature 101, average log power of the Δ frequency band at electrode P3, as the most salient feature. It is quite interesting to see that all three of the top salient EEG features were average log power of

Table 13. Top Four Rankings of Each Saliency Measure

Saliency Measure	Rank 1	Rank 2	Rank 3	Rank 4
Λ_{rank}	$\text{Log}(\Delta)$ at P3 (tie 1)	Number Eye Blinks (tie 1)	Inconclusive	Inconclusive
$\widehat{\Lambda}_{rank}$	$\text{Log}(\Delta)$ at P3	Inconclusive	Inconclusive	Inconclusive
τ_{rank}	$\text{Log}(\Delta)$ at P3	Number Eye Blinks	$\text{Log}(\Delta)$ at FZ	$\text{Log}(\Delta)$ at C4
τ_{rank}^{v1}	$\text{Log}(\Delta)$ at P3	Number Eye Blinks	$\text{Log}(\Delta)$ at FZ	$\text{Log}(\Delta)$ at C4
τ_{rank}^{v2}	$\text{Log}(\Delta)$ at P3	Number Eye Blinks	$\text{Log}(\Delta)$ at FZ	$\text{Log}(\Delta)$ at C4
τ_{rank}^{v3}	$\text{Log}(\Delta)$ at P3	Number Eye Blinks	$\text{Log}(\Delta)$ at FP1	$\text{Log}(\Delta)$ at FZ
SNR_{rank}	$\text{Log}(\Delta)$ at P3	Number Eye Blinks	$\text{Log}(\Delta)$ at FZ	$\text{Log}(\Delta)$ at C4

the Δ frequency band. The number of eye blinks was the only peripheral psychophysiological feature selected as one of the top four salient features.

6.3.5 Conclusions

In summary, several saliency measures were successfully employed to determine the most useful features to classify pilot workload as low or high. The SNR saliency measure appeared to provide saliency rankings consistent with that of partial derivative-based and weight-based saliency measures. Since the weight-based and the SNR saliency measures rely on the sum of the first layer weights squared, it makes sense that they would produce similar results. A set of salient features can be selected from a data set of

Table 14. Four Most Salient Features

Rank	i	Feature
1	101	Average Log Power Of Δ Frequency Band At Electrode P3
2	112	Number Of Eye Blinks
3	98	Average Log Power Of Δ Frequency Band At Electrode FZ
4	100	Average Log Power Of Δ Frequency Band At Electrode C4

EEG and peripheral psychophysiological features using several types of saliency measures.

6.4   *Classifying Air Traffic Controller Workload*

6.4.1   *Introduction*

Before this dissertation research, a set of salient features had not yet been identified for classifying air traffic controller workload. The SNR saliency measure was used to determine the usefulness of psychophysiological features for classifying air traffic controller workload in feedforward MLP ANNs. Thirty-three psychophysiological features were derived from EEG, EOG, ECG, and respiratory gauges in order to classify air traffic controller workload as low, medium, high, or overload. Using the SNR saliency measure, the 33 features were rank ordered. Feature rankings using the SNR saliency measure were statistically shown to be consistent with that of a partial derivative-based saliency measure and a weight-based saliency measure. The SNR saliency measure feature rankings provided a useful way to identify and remove nonsalient features for classifying air traffic controller workload and thus significantly improved the classification accuracy of the validation set.

The objective was to develop a methodology for identifying key psychophysiological features derived from EEG, EOG, ECG, and respiratory gauges for classifying air traffic controller workload. The EEG features used include the power of the five frequency bands in Table 3 collected as the six scalp location shown in Figure 15. The peripheral features used include power of the EOG signal, heart interbeat interval, and respiration interbreath interval. This section is organized in the following

fashion. First, background information to the data collection and preprocessing is summarized. Then, the methodology section describes each step of this air traffic controller workload investigation. Details described in the methodology section include the architecture and training used via a feedforward MLP ANN, the three types of feature saliency measures (a partial derivative-based saliency measure, a weight-based saliency measure, and the SNR saliency measure) used to rank order the features, and the various MOEs using CA. A section contains the results from each step of methodology. In addition, a section provides conclusions.

6.4.2 Data

Data were collected from one fully trained USAF air traffic controller during simulated air traffic control tasks at Los Angeles International Airport using TRACON (Terminal Radar Approach Control), a computer-based air traffic control simulation. EEG is collected at six scalp locations (Fz, Cz, Pz, T5, T6, and O1) shown in Figure 15 following the International 10-20 electrode system. These six scalp locations were selected based upon results from previous studies on air traffic controller workload [15, 177]. EOG, ECG, and respiration measures were also taken.

There were a total of $K = 4$ classes (low, medium, high, and overload) of mental workload. Mental workload levels were selected to correlate well with subjective workload levels using NASA's Task Load Index (TLX) [15, 177]. The low workload condition consisted of controlling six aircraft in 15 minutes. The medium workload condition consisted of controlling 12 aircraft in 15 minutes. The high workload condition consisted of controlling 18 aircraft in 15 minutes. The overload workload condition

consisted of controlling 15 aircraft in five minutes. The test subject was given a three minute break in between random presentations of the workload condition scenarios. Across all four workload conditions, all other factors, such as aircraft type and the ratio of arriving flights to departures and overflights, were held constant. Scenarios were designed so that the workload in each condition increased up to approximately the midpoint of the scenario and then tapered off. EEG, EOG, ECG, and respiratory measures were collected for five minutes at the midpoint of the low, medium, and high workload condition scenario (i.e. minutes 5-10) and for the entire overload condition scenario. The first 30 seconds and the last 30 seconds of data collected were removed. As such, the data collected for the two minutes immediately preceding and for the two minutes immediately following the workload peak for each condition were used.

Thirty EEG power features were developed by passing each EEG signal through a bank of elliptical filters which segmented the signal into five frequency bands as depicted in Table 3 [85]. The power was then calculated for each frequency band over a 10-second moving window with 50% overlap. All EEG data were corrected for eye movements and any portion of the EEG signal that contained other artifacts was discarded.

Three autonomic nervous system features were preprocessed from the EOG, ECG, and respiration gauges. The EOG signal was first passed through a Butterworth lowpass filter with a cutoff frequency of 10 Hertz. The power of the EOG signal was then calculated over a 10-second moving window with 50% overlap. The heart interbeat interval was developed from the ECG signal and was averaged over a 10-second moving window with 50% overlap. The respiration interbreath interval was developed from

respiratory gauges and was, also, averaged over a 10-second moving window with 50% overlap.

A total of 30 EEG features and three autonomic nervous system features were available for input to a feedforward ANN classifier. Three were a total of 47 exemplars for each workload condition. However, one exemplar per workload condition was removed due to artifacts found in the EEG leaving 46 exemplars available for each workload conditions. Since $K = 4$ workload conditions existed with a total of 46 exemplars available for each workload condition, there were a total of $M = 4 \cdot 46 = 184$ exemplars.

6.4.3   Methodology

There were four main steps that included training using all available features, calculating feature saliency, determining if the rankings are positively correlated, and then training all combinations of the ranked features.

6.4.3.1   Step One: Train Using all Available Features

The first step was to train 30 feedforward MLP ANNs as that shown in Figure 3 with a $33 / 66 / 4$ architecture using all 33 features. Each feedforward MLP ANN was trained by the batch error backpropagation algorithm in Equation 50 via *MATLAB*'s *Neural Network Toolbox* with $m_c = 0.90$ and an initial learning rate $\eta = 0.01$. After training for 1000 epochs, the weights for the epoch that produced the minimum SSE_{test} were kept.

Each of the input features were normalized between 0.0 and 1.0 following Equation 19. All hidden and output nodes utilized the sigmoid nonlinear transfer function. Each of the four output nodes corresponded to a workload condition. The desired output vector $\mathbf{d}_m$ for exemplar m was:

$$\mathbf{d}_m = \begin{cases} (0.9 \quad 0.1 \quad 0.1 \quad 0.1) \text{ for } m \in \text{Low Workload Condition} \\ (0.1 \quad 0.9 \quad 0.1 \quad 0.1) \text{ for } m \in \text{Medium Workload Condition} \\ (0.1 \quad 0.1 \quad 0.9 \quad 0.1) \text{ for } m \in \text{High Workload Condition} \\ (0.1 \quad 0.1 \quad 0.1 \quad 0.9) \text{ for } m \in \text{Overload Workload Condition} \end{cases} \qquad (104)$$

Desired output values of 0.1 and 0.9 were used instead of 0.0 and 1.0 to speed up training and to prevent saturation of the sigmoid nonlinear transfer functions. The weights were randomly initialized between -0.5 and 0.5 for each of the 30 training sessions. In addition, the training, test, and validation sets differed for each of the 30 trained feedforward MLP ANNs. For training, 50% of all available exemplars were randomly placed in the *training* set and 25% of all available exemplars in the *test* set [58]. The *validation* set was made up of the remaining 25% of all available exemplars [58]. Since there were a total of $M = 184$ exemplars, $M_{train} = 92$ exemplars were contained in the training set, $M_{test} = 46$ exemplars were contained in the test set, and $M_{valid} = 46$ exemplars were contained in the validation set.

6.4.3.2 Step Two: Calculate Saliency Using Three Types of Saliency Measures

The second step was to train 30 feedforward MLP ANNs using all 33 features plus an injected noise feature with a U(0.0,1.0) distribution in order to calculate the saliency of the 33 features using several saliency measures. The training was conducted in a fashion similar to that described above in Section 6.4.3.1 except that for this second

step, the weights were initialized between -0.1 and 0.1 instead of -0.5 and 0.5 to exploit the fundamental assumption of the weight-based saliency measure and the SNR saliency measure. These measures are based on the assumption that first layer weights emanating from nonsalient features will simply fluctuate around zero during the course of training. As such, it may be desirable to start the weights *close* to zero in order to speed up the training.

Three saliency measures were calculated using a partial derivative-based saliency measure, a weight-based saliency measure, and the SNR saliency measure. The partial derivative-based saliency measure Λ_i for $i = 1,2,...,I = 33$ was computed at the training exemplars following equation 66. Since the sigmoid nonlinear transfer function was used for all activations, then Λ_i for $i = 1,2,...,I = 33$ can be computed specifically following Equation 74 as:

$$\Lambda_i = \frac{1}{K} \cdot \frac{1}{M_{train}} \cdot \sum_{k=1}^{K} \sum_{m=1}^{M_{train}} \left| z_{k,m}(\mathbf{x}'_m, \mathbf{W}) \cdot [1 - z_{k,m}(\mathbf{x}'_m, \mathbf{W})] \cdot \sum_{j=1}^{J} y_{j,m}(\mathbf{x}'_m, \mathbf{W}) \cdot [1 - y_{j,m}(\mathbf{x}'_m, \mathbf{W})] \cdot w_{j,k}^2 \cdot w_{i,j}^1 \right| \quad (105)$$

where $K = 4$ and $M_{train} = 92$. The weight-based saliency measure τ_i for $i = 1,2,...,I = 33$ was computed as the sum of the squared first layer weights emanating from the feature of interest following Equation 77. Finally, the SNR saliency measure SNR_i for $i = 1,2,...,I = 33$ was calculated following Equation 99. The features were then rank ordered by the average over the 30 trained feedforward MLP ANNs for each saliency measure.

158

6.4.3.3 Step Three: Spearman Rank Correlation Tests

For the third step, Spearman rank correlation tests were run to determine if the three types of feature saliency measures, averaged over $G = 30$ training sessions, produced rankings that were consistent. The following ranked pairs were tested using the Spearman rank correlation test:

1. Average partial derivative-based saliency measure rankings versus average weight-based saliency measure rankings

2. Average partial derivative-based saliency measure rankings versus average SNR saliency measure rankings

3. Average weight-based saliency measure rankings versus average SNR saliency measure rankings

The Spearman rank correlation test calculated the correlation between the rankings assigned for each given feature. The null hypothesis H_0 stated that there was no association between the rankings derived from the two average saliency measures. The alternate hypothesis H_a stated that there was a positive correlation between the rankings. A positive correlation implied that the rankings being compared were consistent. The test statistic was calculated as:

$$ r_s = \frac{I \cdot \sum_{i=1}^{I} a_i \cdot b_i - \left(\sum_{i=1}^{I} a_i \right) \cdot \left(\sum_{i=1}^{I} b_i \right)}{\sqrt{\left[I \cdot \sum_{i=1}^{I} a_i^2 - \left(\sum_{i=1}^{n} a_i \right)^2 \right] \cdot \left[I \cdot \sum_{i=1}^{I} b_i^2 - \left(\sum_{i=1}^{I} b_i \right)^2 \right]}} \tag{106} $$

where r_s was the Spearman rank correlation coefficient statistic, and a_i and b_i represented the ranks assigned to feature $i = 1,...,I$ [88]. The test rejected H_0 if $r_s \geq r_{\alpha,I}$ where $r_{\alpha,I}$ is the critical value of the Spearman rank correlation coefficient for a given level of significance α and number of features I. The level of significance was

set to $\alpha = 0.05$ and there were $I = 34$ features. Since $I \geq 30$, the Central Limit Theorem was applicable and so, the sampling distribution of $r_{\alpha,I} = r_{0.05,34}$ was normal [105]. As such, $r_{\alpha,I} = r_{0.05,34} = 0.2818$.

6.4.3.4 Step Four: Train Using Different Combinations of Features

The fourth step utilized the average ranking of the 33 input features derived from the SNR saliency measure. Thirty feedforward MLP ANNs were trained for each combination of the top ranked features. In other words, 30 feedforward MLP ANNs were trained using only the top ranked feature. Next, 30 feedforward MLP ANNs were trained using the top two ranked features. Then, 30 feedforward MLP ANNs were trained using the top three ranked features, and so on. For each training session, the weights were randomly initialized between -0.5 and 0.5. The training, test, and validation sets differed for each of the 30 training sessions.

One sided t – tests were run to determine if the $\overline{CA}$ was significantly increased or decreased as a result of removing nonsalient features for the training, test, and validation sets. In the case of the training set, the null hypothesis H_0 stated that there was no significant difference between the $\overline{CA}_{train}$ with all 33 features and that of the combination that resulted in the highest $\overline{CA}_{valid}$. The alternate hypothesis H_a stated that the $\overline{CA}_{train}$ with all 33 features was lower or higher than that of the combination that resulted in the highest $\overline{CA}_{valid}$. The test statistic assuming the variance of the two samples was unknown and unequal was calculated as:

$$t_s = \frac{\overline{CA}_{train}(33 \text{ features}) - \overline{CA}_{train}\left(\text{Best } \overline{CA}_{valid}\right)}{\sqrt{\dfrac{S_{train}^2(33 \text{ features}) + S_{train}^2\left(\text{Best } \overline{CA}_{valid}\right)}{G}}} \tag{107}$$

where t_s was the $t-$test statistic, $\overline{CA}_{train}(33 \text{ features})$ was the average observed classification accuracy over $G = 30$ trained feedforward MLP ANNs using all 33 features for the training set, $\overline{CA}_{train}\left(\text{Best } \overline{CA}_{valid}\right)$ was the average observed classification accuracy over 30 trained ANNs for the training set using the combination of top ranked features that resulted in the highest $\overline{CA}_{valid}$, $S_{train}^2(33 \text{ features})$ was the sample variance of the CA_{train}^n for $g = 1,...,G$ using all 33 features, and $S_{train}^2\left(\text{Best } \overline{CA}_{valid}\right)$ was the sample variance of the CA_{train}^g for $g = 1,...,G$ using the combination of top ranked features that resulted in the highest $\overline{CA}_{valid}$ [88]. The test rejected H_0 if $|t_s| \geq t_{\alpha,N-1}$ where $t_{\alpha,N-1}$ is the critical t-value for a given level of significance α and degrees of freedom $N-1$. The level of significance was set to $\alpha = 0.05$ and there were $G-1 = 29$ degrees of freedom. As such, $t_{\alpha,N-1} = t_{0.05,29} = 1.6991$. In the case where H_0 was rejected and $t_s < -t_{\alpha,N-1}$, the test concluded that $\overline{CA}_{train}$ with all 33 features was lower than that of the combination that resulted in the highest $\overline{CA}_{valid}$. In the case where H_0 was rejected and $t_s > t_{\alpha,N-1}$, the test concluded that $\overline{CA}_{train}$ with all 33 features was higher than that of the combination that results in the highest $\overline{CA}_{valid}$. Calculations for the test and validation sets were calculated in a fashion similar to Equation 107. In all, there were one $t-$test to compare $\overline{CA}_{train}$, one $t-$test to compare $\overline{CA}_{test}$, and one $t-$test to compare $\overline{CA}_{valid}$.

161

For the combination that resulted in the highest $\overline{CA}_{valid}$, the confusion matrices summed over the 30 training sessions for the training, test, and validation sets were developed. χ^2 tests were then run on each of the rows (hence each true workload condition) of the confusion matrices with all 33 features and with the combination of top ranked features that resulted in the highest $\overline{CA}_{valid}$. (Note that this research was the first to ever perform χ^2 tests on each of the rows of two or more confusion matrices. No methods for statistically comparing two or more confusion matrices were found in the literature.) These χ^2 tests were run for each true workload condition in order to gain insight into whether the proportion of network classification for each true workload condition was altered as a result of removing nonsalient features for the training, test, and validation sets. In the case of the low workload condition for the training set, the null hypothesis H_0 stated that there was no difference between the proportion of network classification for low workload exemplars using all 33 features (Row 1 in the Training Set Confusion Matrix in Table 16) and that using the combination that results in the highest $\overline{CA}_{valid}$ (Row 1 in the Training Set Confusion Matrix in Table 22). The alternate hypothesis H_a stated that the proportion of network classification for low workload exemplars using all 33 features differed from that using the combination that resulted in the highest $\overline{CA}_{valid}$. The test statistic was calculated as:

$$\chi^2_s = 2 \cdot M_{train}^{low} \cdot \left(\sum_{k=1}^{K} \sum_{\ell=1}^{L} \frac{f_{k,\ell}^2}{M_{train}^{low} \cdot \left(\sum_{\ell=1}^{L} f_{k,\ell} \right)} - 1 \right) \tag{108}$$

162

where χ_s^2 is the χ^2 test statistic, M_{train}^{low} is the total number of low workload exemplars in each training set, and $f_{k,\ell}$ is the number of low workload exemplars that the feedforward MLP ANN classified as belonging to class $k = 1,...,K$ in training set confusion matrix $\ell = 1,...,L$ [101]. In this case, $M_{train}^{low} = 713$ and $L = 2$ since two training set confusion matrices were compared. The test rejected H_0 if $\chi_s^2 \geq \chi_{\alpha,K-1}^2$ where $\chi_{\alpha,K-1}^2$ was the critical χ^2 value for a given level of significance α and degrees of freedom $K-1$. The level of significance was set to $\alpha = 0.05$ and there were $K-1 = 3$ degrees of freedom. As such, $\chi_{\alpha,K-1}^2 = \chi_{0.05,3}^2 = 7.8147$. Calculations for comparing the workload conditions in the test and validation confusion matrices were calculated in a fashion similar to Equation 108. In all, 12 (four workload conditions in the training, test, and validation sets) χ^2 tests were run.

6.4.4   Results

6.4.4.1   Step One: Train Using all Available Features

For the first step, 30 feedforward MLP ANNs were trained using all 33 features. Table 15 summarizes several MOEs of various CA forms attained in the training, test, and validation sets. The CIs reported in Table 15 for the expected classification accuracy μ_{CA} had a level of significance set to $\alpha = 0.05$ and $t_{\frac{\alpha}{2},N-1} = t_{\frac{0.05}{2},30-1} = t_{.025,29} = 2.045$. Therefore, the CIs reported provide $100\% \cdot (1-\alpha) = 95\%$ confidence bounds on the estimate of $\overline{CA}$. Of particular interest are the results from the validation set.

163

Table 15. Classification Accuracy Summary Over 30 Trained Feedforward MLP
ANNs Using 33 Features

	Training Set	Test Set	Validation Set
$\overline{CA}$	97.25%	87.39%	84.71%
95% CI for $\overline{CA}$	(96.09%, 98.41%)	(84.95%, 89.83%)	(82.69%, 86.73%)
Min CA, Max CA	88.04%, 100.00%	76.09%, 100.00%	73.91%, 95.65%

Table 16 contains the confusion matrices for the training, test, validation set. Of particular interest is the $\overline{CA}_{valid}$ for the overload condition. Classifying the overload condition correctly is of high importance to the safety of air traffic control. The medium and high workload conditions in the test and validation sets, on average, were not classified as well as the low workload and overload condition. Though the medium and high workload conditions over classified correctly over 90% of the time in the training set, the medium and high workload conditions were classified correctly only over 70% of the time in the test and validation sets.

6.4.4.2 Step Two: Calculate Saliency Using Three Types of Saliency Measures

The second step calculated the average saliency of the 33 features over 30 trained ANNs for all three saliency measures. Table 17 lists the features for each average saliency rankings over 30 trained feedforward MLP ANNs. Table 18 lists the average saliency ranking for each feature. The information provided in Table 17 is the same as that provided in Table 18 but in a different format.

Table 16. Confusion Matrices Summed Over 30 Trained ANNs Using 33 Features

		Training Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	702 98.46%	1 0.14%	10 1.40%	0 0.00%	713
	Medium	7 1.00%	657 94.26%	33 4.73%	0 0.00%	697
	High	4 0.59%	21 3.11%	650 96.30%	0 0.00%	675
	Overload	0 0.00%	0 0.00%	0 0.00%	675 100.00%	675
	Overall	713	679	693	675	2760 97.25%

		Test Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	330 95.10%	16 4.61%	1 0.29%	0 0.00%	347
	Medium	17 4.99%	270 79.18%	53 15.54%	1 0.29%	341
	High	20 5.57%	58 16.16%	278 77.44%	3 0.84%	359
	Overload	0 0.00%	0 0.00%	5 1.50%	328 98.50%	333
	Overall	367	344	337	332	1380 87.39%

		Validation Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	302 94.38%	11 3.44%	7 2.19%	0 0.00%	320
	Medium	19 5.56%	248 72.51%	74 21.64%	1 0.29%	342
	High	23 6.65%	69 19.94%	249 71.97%	5 1.45%	346
	Overload	0 0.00%	0 0.00%	2 0.54%	370 99.46%	372
	Overall	344	328	332	376	1380 84.71%

Table 17. Features for Each Average Saliency Rankings Over 30 Trained Feedforward MLP ANNs

Average Ranking	Partial Derivative-Based Saliency	Weight-Based Saliency	SNR Saliency
1	β Power at T5	$\mu\beta$ Power at T6	$\mu\beta$ Power at T6
2	$\mu\beta$ Power at O1	$\mu\beta$ Power at O1	β Power at T5
3	$\mu\beta$ Power at T5	β Power at T5	$\mu\beta$ Power at O1
4	$\mu\beta$ Power at T6	$\mu\beta$ Power at T5	$\mu\beta$ Power at T5
5	θ Power at Fz	β Power at T6	β Power at T6
6	Interbreath Interval	θ Power at Fz	Interbreath Interval
7	$\mu\beta$ Power at Cz	Interbreath Interval	Interbeat Interval
8	Interbeat Interval	Interbeat Interval	θ Power at Fz
9	β Power at T6	$\mu\beta$ Power at Cz	$\mu\beta$ Power at Cz
10	θ Power at Pz	θ Power at Pz	α Power at O1
11	α Power at O1	α Power at O1	θ Power at Pz
12	θ Power at T5	Δ Power at Pz	θ Power at T5
13	Δ Power at Pz	θ Power at T5	α Power at T6
14	α Power at T6	Δ Power at Fz	Δ Power at Pz
15	β Power at Fz	α Power at T6	Δ Power at Fz
16	Δ Power at Fz	β Power at Fz	Eye Power
17	α Power at Pz	Eye Power	β Power at Fz
18	α Power at Fz	α Power at Fz	α Power at Pz
19	Eye Power	α Power at Pz	β Power at O1
20	Δ Power at T6	Δ Power at T6	α Power at Fz
21	Δ Power at T5	Δ Power at T5	θ Power at T6
22	θ Power at Cz	Δ Power at Cz	θ Power at Cz
23	Δ Power at Cz	θ Power at Cz	Δ Power at T5
24	$\mu\beta$ Power at Pz	β Power at O1	α Power at Cz
25	θ Power at T6	$\mu\beta$ Power at Pz	Δ Power at T6
26	α Power at T5	θ Power at T6	α Power at T5
27	α Power at Cz	α Power at T5	Δ Power at Cz
28	β Power at O1	α Power at Cz	$\mu\beta$ Power at Pz
29	β Power at Cz	$\mu\beta$ Power at Fz	Δ Power at O1
30	β Power at Pz	Δ Power at O1	$\mu\beta$ Power at Fz
31	Δ Power at O1	β Power at Cz	β Power at Cz
32	$\mu\beta$ Power at Fz	β Power at Pz	θ Power at O1
33	θ Power at O1	θ Power at O1	β Power at Pz
34	Injected Noise	Injected Noise	Injected Noise

166

Table 18. Average Saliency Rankings for Each Feature Over 30
Trained Feedforward MLP ANNs

Feature i	$\overline{\Lambda}_i$ Ranking	τ_i Ranking	SNR_i Ranking
Δ Power at Fz	16	14	15
Δ Power at Cz	23	22	27
Δ Power at T5	21	21	23
Δ Power at Pz	13	12	14
Δ Power at T6	20	20	25
Δ Power at 01	31	30	29
θ Power at Fz	5	6	8
θ Power at Cz	22	23	22
θ Power at T5	12	13	12
θ Power at Pz	10	10	11
θ Power at T6	25	26	21
θ Power at 01	33	33	32
α Power at Fz	18	18	20
α Power at Cz	27	28	24
α Power at T5	26	27	26
α Power at Pz	17	19	18
α Power at T6	14	15	13
α Power at 01	11	11	10
β Power at Fz	15	16	17
β Power at Cz	29	31	31
β Power at T5	1	3	2
β Power at Pz	30	32	33
β Power at T6	9	5	5
β Power at 01	28	24	19
$\mu\beta$ Power at Fz	32	29	30
$\mu\beta$ Power at Cz	7	9	9
$\mu\beta$ Power at T5	3	4	4
$\mu\beta$ Power at Pz	24	25	28
$\mu\beta$ Power at T6	4	1	1
$\mu\beta$ Power at 01	2	2	3
Power of EOG	19	17	16
Interbreath Interval	8	8	7
Interbeat Interval	6	7	6
Injected Noise	34	34	34

Regardless of saliency measure, the injected noise feature was always the least salient feature. Of the autonomic nervous system features, respiration interbreath interval was, on average, the most salient feature. The heart interbeat interval was, on average, the second most salient autonomic nervous system feature and power of the EOG signal was, on average, the least salient autonomic nervous system feature. For EEG, those features derived from the $\mu\beta$ frequency band appeared, on average, to be the most salient. In fact, three of the top four most salient features were derived from the $\mu\beta$ frequency band for all three saliency measures. Those features derived from the Δ frequency band appeared, on average, to be the least salient.

6.4.4.3 Step Three: Spearman Rank Correlation Tests

The Spearman rank correlation test was performed in the third step. Table 19 summarizes the results from the three Spearman rank correlation tests performed. In all cases, H_0 was rejected and the test concluded that all three saliency measures produced rankings that were, on average, consistent with 95% confidence.

Table 19. Results from the Spearman Rank Correlation Tests

Saliency Measures Compared	r_s	$r_s \geq 0.282$	Conclusion
Partial derivative-based saliency measure to weight-based saliency measure	0.9858	Yes	Reject H_0
Partial derivative-based saliency measure to SNR saliency measure	0.9612	Yes	Reject H_0
Weight-based saliency measure to SNR saliency measure	0.9752	Yes	Reject H_0

6.4.4.4 Step Four: Train Using Different Combinations of Features

In the fourth step, 30 training sessions were performed for each combination of the top ranked features. Figure 34 contains plot of the $\overline{CA}$, the 95% CI for μ_{CA}, the minimum classification accuracy, and the maximum classification accuracy for the training, test, and validation sets. A very large dip occurs in the minimum classification accuracy curves in the training and validation sets with the top 31 ranked features. This dip is more than likely a result of the error backpropagation training algorithm getting stuck in a local minimum of the error-weight space.

The results for the validation set are of particular interest. From the bottom plot of Figure 34, the highest $\overline{CA}_{valid}$ corresponds to the feature set combination that contains the top 17 ranked features. Table 20 summarizes the classification accuracies attained in the training, test, and validation sets using the top 17 ranked features. The classification accuracy results with the top 17 ranked features in Table 20 can be directly compared to that using all 33 features in Table 15. Table 21 summarizes the results from the three t-tests performed. Both $\overline{CA}_{train}$ and $\overline{CA}_{test}$ did not significantly increase or decrease after removing 16 nonsalient features with 95% confidence.

Table 20. Classification Accuracy Summary Over 30 Trained Feedorward MLP
ANNs Using Top 17 Ranked Features

	Training Set	Test Set	Validation Set
$\overline{CA}$	97.17%	89.78%	87.10%
95% CI for $\overline{CA}$	(96.03%, 98.32%)	(87.80%, 91.76%)	(85.16%, 89.04%)
Min CA, Max CA	90.22%, 100.00%	73.91%, 100.00%	76.09%, 97.83%
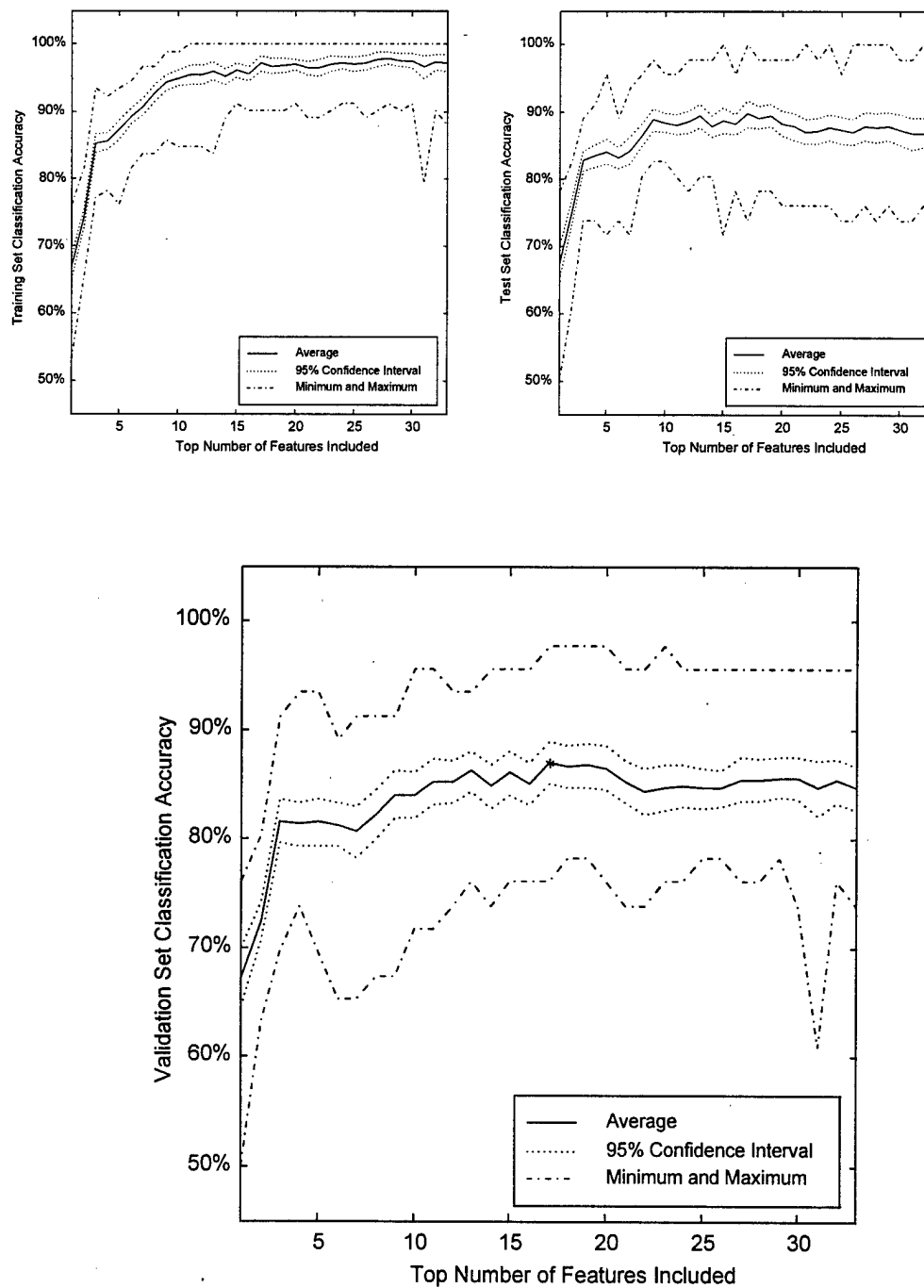
Figure 34. $\overline{CA}$ Over 30 Trained Feedforward MLP ANNs (Note: Feature Rankings Selected by SNR Saliency Measure)

170

Table 21. Results from t – Tests

$\overline{CA}$ Compared	t_s	$|t_s| \geq 1.6991$	Conclusion
$\overline{CA}_{train}$ (33 features) versus $\overline{CA}_{train}$ (17 features)	0.0908	No	Fail to reject H_0, $\overline{CA}_{train}$ (33 features) $= \overline{CA}_{train}$ (17 features)
$\overline{CA}_{test}$ (33 features) versus $\overline{CA}_{test}$ (17 features)	-1.5555	No	Fail to reject H_0, $\overline{CA}_{test}$ (33 features) $= \overline{CA}_{test}$ (17 features)
$\overline{CA}_{valid}$ (33 features) versus $\overline{CA}_{valid}$ (17 features)	-1.7472	Yes	Reject H_0, $\overline{CA}_{valid}$ (33 features) $< \overline{CA}_{valid}$ (17 features)

However, H_0 was rejected and the test concluded that $\overline{CA}_{valid}$ significantly increased after removing 16 nonsalient features with 95% confidence.

Table 22 contains the confusion matrices for the training, test, and validation sets, respectively, using the top 17 ranked features. The rows of confusion matrices using the top 17 ranked features can be compared to that of the confusion matrices using all 33 features in Table 16.

The $\overline{CA}_{valid}$ for the overload condition in Table 22 is of particular interest. As in the case with all 33 features, the $\overline{CA}_{test}$ and $\overline{CA}_{valid}$ for the medium and high workload conditions were, on average, lower that than of the low workload and overload conditions. Though the $\overline{CA}_{train}$ was greater than 90% for the medium and high workload conditions, the $\overline{CA}_{test}$ and $\overline{CA}_{valid}$ were only greater than 75% for the medium and high workload conditions. By removing 16 nonsalient features, the classification of the high workload condition in the validation set was significantly improved. The change in the medium workload condition in the validation set was, however, not significantly changed.

Table 23 summarizes the results from the 12 χ^2 tests performed. In all but one comparison, the proportion of network classification for each workload condition did not significantly change as a result of removing 16 nonsalient features in the training, test, and validation sets. The only χ^2 test where H_0 was rejected occurred when comparing the proportion of network classification of the high workload condition in the validation set. Since $\overline{CA}_{valid}$ significantly increased after removing 16 nonsalient features (see Table 21), at least one workload condition in the validation set was expected to change significantly.

6.4.5 Conclusions

The first step showed that features from EEG, EOG, ECG, and respiratory gauges show potential for classifying air traffic controller workload as low, medium, high, and overload. The $\overline{CA}_{valid}$ over 30 training sessions was 84.71% using 33 features derived from EEG, EOG, ECG, and respiratory gauges via a feedforward MLP ANN. In addition, the $\overline{CA}_{valid}$ over 30 training sessions for the overload condition using 33 features was 99.46%.

In the second step, a noise feature was injected and several types of saliency measures were calculated for all 33 features. The saliency measures computed included the partial derivative saliency measure, the weight-based saliency measure, and the SNR saliency measure. From these three measures, the features were rank ordered.

In the third step, the rankings provided by the three types of saliency measures were shown to be, on average, consistent with 95% confidence. There is a computational

Table 22. Confusion Matrices Summed Over 30 Trained ANNs Using 17 Features

		Training Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	704 98.74%	2 0.28%	7 0.98%	0 0.00%	713
	Medium	7 1.00%	655 93.97%	35 5.02%	0 0.00%	697
	High	4 0.59%	23 3.41%	648 96.00%	0 0.00%	675
	Overload	0 0.00%	0 0.00%	0 0.00%	675 100.00%	675
	Overall	715	680	690	675	2760 97.17%

		Test Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	324 93.37%	13 3.75%	10 2.88%	0 0.00%	347
	Medium	15 4.40%	285 83.58%	41 12.02%	0 0.00%	341
	High	15 4.18%	43 11.98%	300 83.57%	1 0.28%	359
	Overload	0 0.00%	0 0.00%	3 0.90%	330 99.10%	333
	Overall	367	344	337	332	1380 89.78%

		Validation Set Network Classification				
		Low	Medium	High	Overload	Overall
True Classification	Low	301 94.06%	12 3.75%	7 2.19%	0 0.00%	320
	Medium	19 5.56%	259 75.73%	64 18.71%	0 0.00%	342
	High	22 6.36%	53 15.32%	271 78.32%	0 0.00%	346
	Overload	0 0.00%	0 0.00%	1 0.27%	371 99.73%	372
	Overall	342	324	343	371	1380 87.10%

Table 23. Results from χ^2 Tests Comparing Rows of Confusion Matrices with 33
Features to that with Top 17 Ranked Features

	True Workload	χ_s^2	$\chi_s^2 \geq 7.8147$	Conclusion
Training **Set**	Low	0.8656	No	Fail to reject H_0, Equal network classification proportions
	Medium	0.0619	No	Fail to reject H_0, Equal network classification proportions
	High	0.0940	No	Fail to reject H_0, Equal network classification proportions
	Overload	0.0000	No	Fail to reject H_0, Equal network classification proportions
Test **Set**	Low	7.7290	No	Fail to reject H_0, Equal network classification proportions
	Medium	3.0623	No	Fail to reject H_0, Equal network classification proportions
	High	4.7794	No	Fail to reject H_0, Equal network classification proportions
	Overload	0.5061	No	Fail to reject H_0, Equal network classification proportions
Validation **Set**	Low	0.0451	No	Fail to reject H_0, Equal network classification proportions
	Medium	1.9633	No	Fail to reject H_0, Equal network classification proportions
	High	8.0514	Yes	Reject H_0, Network classification proportions not equal
	Overload	0.3347	No	Fail to reject H_0, Equal network classification proportions

advantage to using the weight-based saliency measure and the SNR saliency measure over the partial derivative-based saliency measure. Both the weight-based saliency measure and the SNR saliency measure take only a fraction of the time it takes to calculate the partial derivative-based saliency measure. The SNR saliency measure also offers the advantage of comparing the saliency of each feature to that of an injected known noisy nonsalient feature.

Finally, statistical tests showed in the fourth step that the removal of 16 nonsalient features did, in no way, decrease any classification capability of a feedforward MLP ANN to classify air traffic control workload. In fact, the $\overline{CA}_{valid}$ was significantly increased by removing nonsalient features. Using the top 17 ranked features produced a $\overline{CA}_{valid}$ over 30 training sessions of 87.10%. Thus, 48.48% of the features can be removed while actually improving the $\overline{CA}_{valid}$ by 2.39% which was a significant increase with 95% confidence. The $\overline{CA}_{valid}$ over 30 training sessions for the overload condition using the top 17 ranked features increased slightly to 99.73%, though this was not a significant increase. The increase in $\overline{CA}_{valid}$ was mainly attributed to the improvement in classification of the high workload condition.

Of the top 17 features, all three of the of autonomic nervous system features were included. This showed that peripheral psychophysiological features are useful, as expected, for classifying air traffic controller workload. Of the EEG frequency bands, features from the $\mu\beta$ frequency band were selected more often than any other frequency band. Selection of features from the $\mu\beta$ frequency band agreed with stepwise discriminant analysis results and principle component analysis results from Wilson and Fisher's mental workload study in which subjects performed 14 cognitive tasks [174].

However, questions loom about the biological insight to the selection of features from the $\mu\beta$ frequency band. The physiological origin of the of the $\mu\beta$ frequency band is of question since it is not known whether the higher frequencies are from the brain or from the musculature covering the scalp [174]. Regardless of the physiological origin of the $\mu\beta$ frequency band, it is clear that the features derived from the $\mu\beta$ frequency band for this investigation were, on average, the most salient. Finally, selection of EEG features mostly from the scalp locations of Fz, T5, and T6 showed that air traffic control involves many areas of the brain. This was expected since air traffic control is a complex and demanding task.

6.5   Conclusions

In conclusion, the methodology as developed for classifying the workload of a pilot in addition to that of an air traffic controller appears to be able to remove nonsalient features while maintaining, and in some cases improving, classification of air traffic controller workload. Feature saliency is an important step in developing any ANN model for improving generalization capability.

7   Signal-to-Noise Ratio (SNR) Screening Method as Applied to Classifying Pilot Workload via Feedforward Multilayer (MLP) Artificial Neural Networks (ANN) in Addition to Elman Recurrent Neural Networks (RNN)

7.1   Introduction

The SNR screening method is a new feature screening technique. The SNR screening method utilizes the SNR saliency measure in order to produce a parsimonious set of salient features while maintaining good classification accuracy. As with the SNR saliency measure, Bauer proposed the SNR screening method [6] and Sumrell was the first to experiment with the SNR screening method using a noisy version of the XOR classification problem in Figure 2 and Fisher's iris classification problem [147]. The SNR screening method significantly reduces the number of features inputted to an ANN while maintaining classification accuracy or, in some cases, significantly improving classification accuracy. The SNR screening method may account for feature redundancy. The SNR screening method holds the potential to require only a single training run to remove all nonsalient features.

This chapter summarizes the application of the SNR screening method to modeling pilot workload via feedforward MLP ANNs in addition to Elman RNNs as published in [47] and to appear in [49]. This dissertation research produced the first non-trivial, real-world applications of the SNR screening method in both feedforward MLP ANNs and Elman RNNs.

The objective of both the feedforward MLP ANN and the Elman RNN was to estimate a pilot's workload while landing an airplane in a similar fashion to that as

described in Section 6.3. This chapter will first described the SNR screening method. Next, the application of the SNR screening method to a feedforward MLP ANN for *classifying* a pilot's workload while landing an airplane is summarized. Then, the application of the SNR screening method to an Elman RNN for *estimating* a pilot's workload while landing an airplane is summarized.

7.2 Signal-to-Noise Ratio (SNR) Screening Method

The SNR screening method is a backwards screening method that utilizes the SNR saliency measure. Previously developed feature screening methods, such as the Belue-Bauer screening method [11, 12] discussed in Section 3.5.2 and the Steppe-Bauer screening method [136, 137, 140] discussed in Section 3.5.3, utilize a partial derivative-based saliency measure [124, 126] or a weight-based saliency measure [152]. The SNR screening method holds the promise of potentially identifying and removing nonsalient input features in a single training run. Both the Belue-Bauer and the Steppe-Bauer screening method typically require between 10 to 30 training runs [11, 12, 136, 137, 140]. The SNR saliency measure appears highly robust relative to the effects of the weight initialization, the ANN's architecture, and the selection of training and test sets.

The SNR screening method provides a mechanism to potentially identify a parsimonious set of salient features by removing non-salient features while striving to maintain good generalization.

Signal-to-Noise Ratio (SNR) Screening Method

1. Introduce a Uniform (0,1) noise feature x_N to the original set of features.

2. Normalize all features following Equation 16 or Equation 19.

178

3. Randomly initialize the weights between -0.001 and 0.001.

4. Randomly select the training and test sets.

5. Begin to train the ANN.

6. After each epoch, compute the SNR saliency measure for each input feature.

7. Interrupt training when the SNR saliency measures for all input features have stabilized.

8. Compute CA_{test}.

9. Identify the feature with the lowest SNR saliency measure and remove it from further training.

10. Continue training the ANN.

11. Repeat steps 6 through 9 until all the features (except the noise feature) in the original set are removed from training.

12. Compute the reaction of CA_{test} due to the removal of the individual features.

13. Retain the first feature whose removal caused a significant decrease in CA_{test}, as well as all features which were removed after that first salient feature.

14. Retrain the ANN with only the parsimonious set of salient input features.

Because the SNR saliency measure directly compares the saliency of each feature to a baseline noise feature, the SNR saliency measure has the potential to be used "on the fly" to rank order the features while the ANN is training. In other words, the SNR screening method, which utilizes the SNR saliency measure, may be completed in only a single training run of the ANN. The potential to remove all non-salient features in a single run would be a distinct advantage over the Belue-Bauer screening method and the Steppe-Bauer screening method which typically require between 10 to 30 runs [11, 12, 136, 137, 140].

7.3 Signal-to-Noise Ratio (SNR) Screening Method in Feedforward Multilayer (MLP) Artificial Neural Networks (ANN)

7.3.1 Introduction

The SNR screening method was applied to the same pilot workload classification problem described in Section 6.3 to assess its potential to remove nonsalient features within a single training run of a feedforward MLP ANN. Results from Section 6.3 determined that the four most salient features using the SNR saliency measure included number of eye blinks and the average log power of the Δ frequency band at electrodes P3, Fz, and C4. Because three of the top four salient features were log power of the Δ frequency band at three different electrodes, this may suggest that the SNR saliency measure does not account for feature redundancy. The SNR screening method may account for redundant features.

7.3.2 Data

The same pilot workload classification data set with 18 prescreened features as described in Section 6.3 was used to explore the ability of the SNR screening method to identify and remove nonsalient features. Seventeen features of the average log power of the EEG signals in addition to the number of eye blinks over a 10-second moving window with 50% overlap served as inputs to a feedforward MLP ANN to classify pilot workload as low or high.

7.3.3 Methodology

The SNR screening method was replicated 20 times, each time with a different random seed. As such, each replication had a different set of initialized weights between -0.001 and 0.001. Also, each replication randomly selected two-thirds of the data for training and one-third of the data for testing. A noise feature with a Uniform(0.0,1.0) distribution was injected. All features were normalized following Equation 19. Each feedforward MLP ANN trained had a 19/19/2 architecture. All hidden and output nodes were activated by the sigmoid nonlinear transfer function in Equation 4. A fixed learning rate $\eta = 0.3$ was used. No momentum was used. Training of each feedforward MLP ANN was stopped when all of the following stabilized:

- MSE_{train}
- MSE_{test}
- $\mathbf{W}$
- $SNR_i \; \forall \; i = 1,2,\ldots,I$.

7.3.4 Results

The SNR screening method, on average, took 2747.6 epochs to remove all features and identify a parsimonious set of salient features. Seventeen out of 20 times (85%), the SNR screening method selected average log power of the Δ frequency band at electrode P3, as the only feature required in order to attain $CA_{test} = 100\%$. Three out of 20 times (15%), the SNR screening method determined that average log power of the Δ frequency band at electrode P3 and number of eye blinks were the only features required to maintain $CA_{test} = 100\%$. $CA_{test} = 100\%$ is an improvement over $CA_{test} = 90\%$ using the top four salient features from Section 6.3.

Figure 35. $\overline{CA}_{test}$ Resulting From Feature Removal
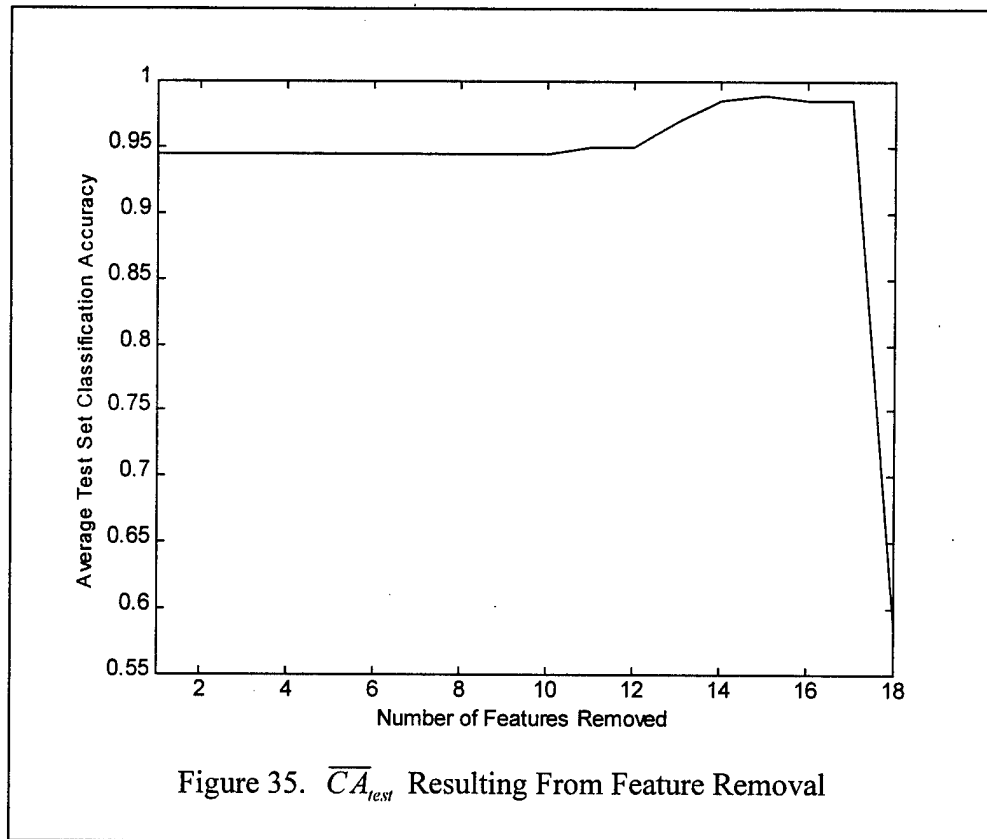
Figure 35 plots $\overline{CA}_{test}$ as a result of feature removal over the 20 independent training sessions. After the first feature was removed, $\overline{CA}_{test} = 94.5\%$. In fact, $\overline{CA}_{test}$ remained at 94.5% until 11 features were removed. $\overline{CA}_{test}$ gradually increased each time an additional feature was removed until 16 features were removed. $\overline{CA}_{test}$ then slightly increased after 16 features were removed and remained the same after 17 features were removed. Finally, the $\overline{CA}_{test}$ decreased significantly after 18 features were removed. Figure 35 suggests that $\overline{CA}_{test}$ may increase as nonsalient features are removed.

7.3.5 Conclusions

In addition to the parsimonious set of salient features improving the CA_{test}, the selected set of salient features for classifying pilot workload is physiologically sound.

Landing an aircraft is a very demanding visual task. Number of eyes blinks is inversely related to the level of difficulty of a visual task and has been shown to be a sensitive measure to visual workload [15, 170, 173, 175, 177]. The more difficult a visual task, the less a human will blink [15, 170, 173, 175, 177] (see Section 4.3.1.1 for a discussion of EOG for classifying mental workload). As such, the selection of the number of eye blinks in three of the 20 replications makes sense. Also, the selection of the average log power of the Δ frequency band at electrode P3 makes sense. The electrode P3 as shown in Figure 16 is located at the parietal area of the brain (left side of the back of the head) and is commonly known as the "cortical association area" which performs both reasoning and secondary processing of sensory information [100]. In addition, Harmony et al. showed that the EEG Δ frequency band is an indicator of attention to internal processing during performance of mental tasks [54].

Not once was the average power of the EEG Δ band at two different scalp locations selected for the parsimonious set of salient features. This suggests that the SNR screening method may help account for redundant features.

7.4 Signal-to-Noise Ratio (SNR) Screening Method in Elman Recurrent Neural Networks (RNN)

7.4.1 Introduction

EEG continually changes in both time and space. Feedforward MLP ANNs as utilized in previous chapters of this dissertation can adequately reflect the differences and changes of EEG over the electrodes as shown in Figure 15 and Figure 16. Feedforward

MLP ANNs can also adequately reflect differences and changes of EEG over the various frequency bands as listed Table 3 and Table 4. However, feedforward MLP ANNs can not reflect the changes in EEG in addition to the peripheral psychophysiological features over time. Since there is a strong temporal component to EEG and to the peripheral psychophysiological features, the next logical step was to model pilot workload using a type of ANN that allows for the encoding of time such as the Elman RNN [31] as depicted in Figure 38. Whereas the pilot workload research described in Chapter 6 and in Section 7.3 *classified* workload as low or high, the pilot workload summarized here *estimates* pilot workload.

7.4.2 Data

Data similar to that described in Section 6.3.2 were used. Features were derived from EEG and peripheral psychophysiological measures collected while a pilot flew simulated landing scenarios on two different days. Data collected on the first day was used as training data. Data collected on the second day was used as test data. (The data described in Section 6.3.2 was collected in one day). The landing scenario was repeated five times on each data collection day.

Both the training set and the test set contained a total of 112 psychophysiological features (108 EEG features and four physiological features). A description of these 112 psychophysiological features is given in Section 6.3.2. The training set contained only 32 exemplars. The test set also contained only 32 exemplars. Figure 36 contains plots of a few of the input features before normalization.
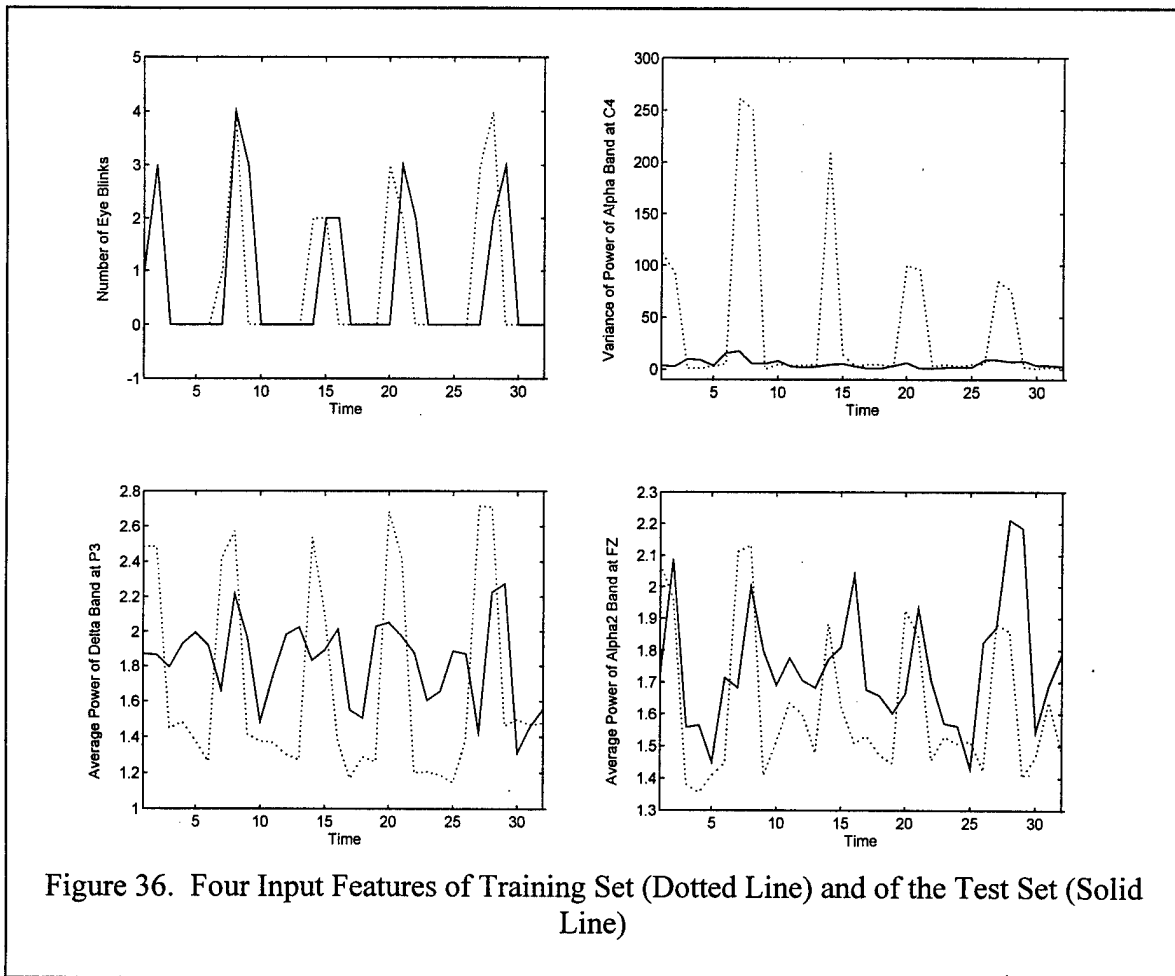
Figure 36. Four Input Features of Training Set (Dotted Line) and of the Test Set (Solid Line)

Whereas the pilot workload research described in Chapter 6 and in Section 7.3 *classified* workload as low or high, now levels of pilot workload range between 0.0 and 1.0. When the pilot is flying straight and level, his workload is low and is therefore given a workload value of 0.0. When the pilot begins his descent to the airfield, his workload is assumed to increase linearly until touchdown. For this scenario, a pilot's workload is highest at touchdown and is therefore given a workload value of 1.0. After each touchdown, the pilot was given a 30-second rest period and then the scenario was reset. Data from the 30-second rest period was removed and not considered for analysis. Figure 37 is a plot of the desired output of the training set and of the test set. Note from

185
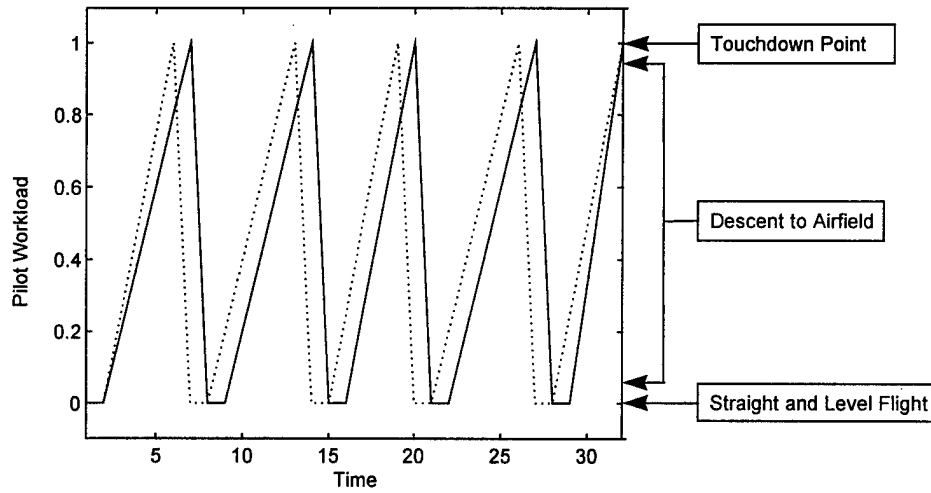
Figure 37. Desired Output of Training Set (Dotted Line) and of Test Set (Solid Line)

Figure 37 that the time it takes to land the airplane differs between the training set and the test set. Also note from Figure 37 that the time it takes to land the airplane differs between each landing scenario. In some cases, it only took the pilot three time periods to land the airplane. In other cases, it took the pilot five time periods to land the airplane. The amount of time it takes to land an airplane can depend on several factors such as airspeed, rate of descent, and touchdown point.

7.4.3 Methodology

7.4.3.1 Step One: Train Using all Candidate Input Features over Experimental Design

The first step was to train Elman RNNs [31] with a $112 + J / J / 1$ architecture, as depicted in Figure 38, using all 112 available features over a 5^2 full-factorial experimental design. A 5^2 full-factorial experimental design was used to examine the effects of two factors, m_c and J, on $RMSE_{test}$ as in Equation 48. Five values of m_c were evaluated: 0.1, 0.3, 0.5, 0.7, and 0.9. Five values of J were evaluated: 1, 3, 5, 7, and 9.

Figure 38. $112 + J / J / 1$ Elman RNN
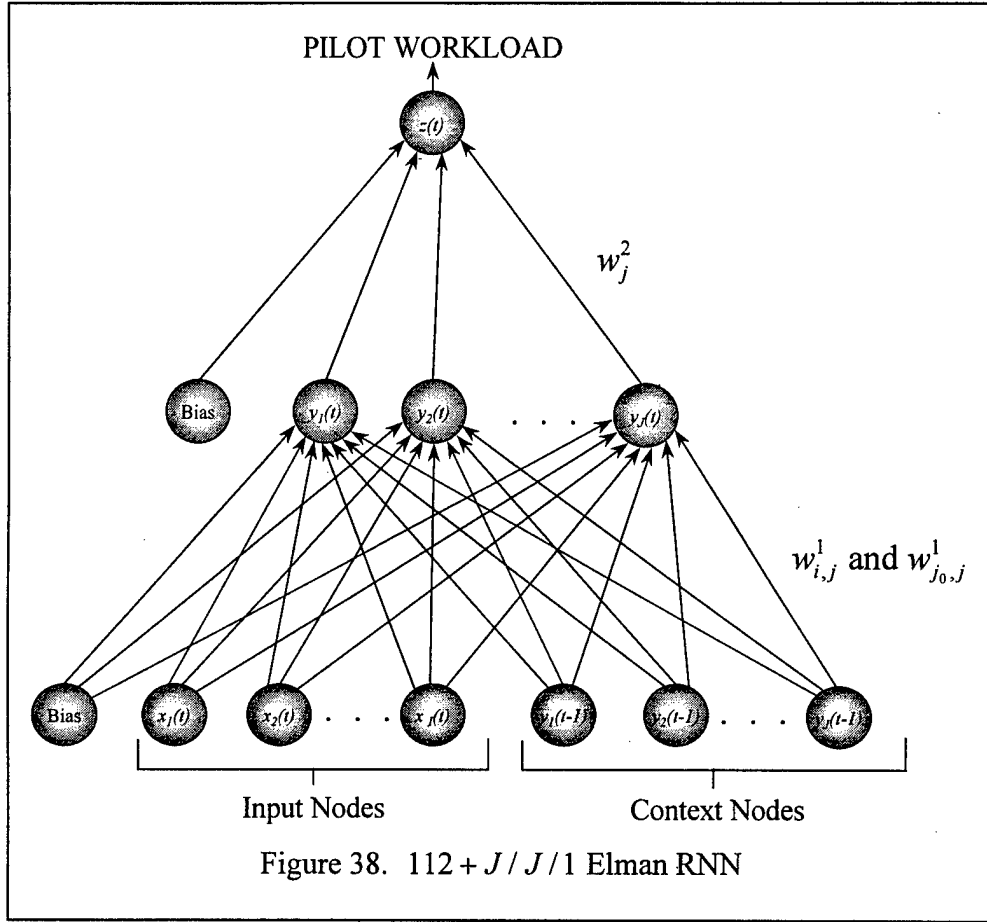
Cover's Theorem as extended to hidden nodes in Equation 61 provided guidance in the selection of the levels of J evaluated [23]. Ten replications were performed at each of the 25 (5^2) design points resulting in a total of 250 trained Elman RNNs using all 112 features. All hidden/context node were activated by the hyperbolic tangent nonlinear transfer in Equation 6. A linear transfer function with *slope* = 1 in Equation 8 was used on the output node.

Using the *Matlab Neural Network Toolbox*, each of the 250 Elman RNNs was trained using the error backpropagation algorithm with an adaptive learning rate η and momentum in Equation 50 until $RMSE_{train} < 0.02$ or for $E = 2000$ epochs, whichever occurred first. The set of weights for the epoch that produced the minimum $RMSE_{test}$

187

was kept. For each design point, the average minimum root mean squared error of the test set denoted as $\overline{RMSE_{test}}$ was calculated in addition to the average number of epochs denoted as $\overline{E}$ required to reach the minimum $RMSE_{test}$.

7.4.3.2 Step Two: Perform Signal-to-Noise Ratio (SNR) Screening Method over Experimental Design

The next step was to perform the SNR screening method over the 5^2 full-factorial experimental design in order to determine the parsimonious set of salient features for estimating a pilot's workload while landing an airplane. Since there are only 32 exemplars, training an Elman RNN with 112 input features is in violation of Foley's Rule [37] in Equation 59 and the so-called "curse of dimensionality" [28] is clearly apparent. The SNR screening method may cure this data set's "curse of dimensionality" by eliminating enough nonsalient features. The SNR screening method may reduce the $RMSE_{test}$. This was the first attempt ever to apply the SNR screening method to an Elman RNN.

<u>Modified Signal-to-Noise Ratio (SNR) Screening Method</u>

1. Inject a noise feature x_N with a Uniform(0.0,1.0) distribution to the original set of features.

2. Normalize all features between -1.0 and 1.0 using Equation 20.

3. Randomly initialize the weights between -0.001 and 0.001.

4. Begin training the Elman RNN.

5. After every epoch, compute $RMSE_{train}$ and $RMSE_{test}$.

6. Stop training when $RMSE_{train} < 0.02$ or after 500 epochs, whichever happens first.

7. Compute the SNR saliency measure following Equation 99 for each input feature. Note that the SNR saliency measure is not computed for the context nodes.

8. Identify the feature with the lowest SNR saliency measure and remove it from further training.

9. Continue training the Elman RNN.

10. After every epoch, compute $RMSE_{train}$ and $RMSE_{test}$.

11. Stop training when $RMSE_{train} < 0.02$ or after 100 epochs, whichever happens first.

12. Compute the SNR saliency measure for each input feature.

13. Identify the feature with the lowest SNR saliency measure and remove it from further training.

14. Repeat steps 9 through 13 until all the features (except the noise feature) in the original set are removed from training.

15. Determine when the minimum $RMSE_{test}$ occurred during training.

16. Retain all features that were removed after the minimum $RMSE_{test}$ occurred.

The same 5^2 full-factorial design of experiments was used with 10 replications at each design point. As such, the SNR screening method was performed 250 times.

7.4.3.3 Step Three: Train Using Parsimonious Set of Salient Features over Experimental Design

The final step in this preliminary investigation was to train Elman RNNs using the parsimonious set of salient features as determined by the SNR screening method over the 5^2 full-factorial experimental design. Once again, 10 replications were performed at each design point resulting in a total of 250 Elman RNNs trained on the parsimonious set

of salient features. As in Step One in Section 7.4.3.1, each of the 250 Elman RNNs were trained with the *Matlab Neural Network Toolbox* using the error backpropagation algorithm with an adaptive learning rate η and momentum in Equation 50 until $RMSE_{train} < 0.02$ or for $E = 2000$ epochs, whichever occurred first. The set of weights for the epoch that produced the minimum $RMSE_{test}$ was kept. For each design point, the average minimum $\overline{RMSE}_{test}$ was calculated in addition to $\overline{E}$ required to reach the minimum $RMSE_{test}$.

7.4.4   Results

7.4.4.1   Step One: Train Using all Candidate Input Features over Experimental Design

The first step was to train 250 Elman RNNs following the experimental design. Figure 39 is a plot of the $RMSE_{train}$ and the $RMSE_{test}$ for one of the training sessions with $m_c = 0.1$ and $J = 3$. For the training session depicted in Figure 39, the minimum $RMSE_{test} = 0.2874$ occurred after 95 epochs of training. Training was stopped after 1521 epochs when $RMSE_{train} < 0.02$.

Table 24 provides a summary of the minimum $\overline{RMSE}_{test}$ using 112 features. The design point that produced the best $\overline{RMSE}_{test}$ is shaded in Table 24. The best $\overline{RMSE}_{test} = 0.2893$ was achieved with $m_c = 0.1$ and $J = 3$. But upon closer examination of Table 24, it becomes apparent that the selection of m_c and J did not significantly affect $\overline{RMSE}_{test}$. Overall, $\overline{RMSE}_{test} = 0.2930$.

Figure 39. $RMSE_{train}$ (Dotted Line) and $RMSE_{test}$ (Solid Line) with 112 Features
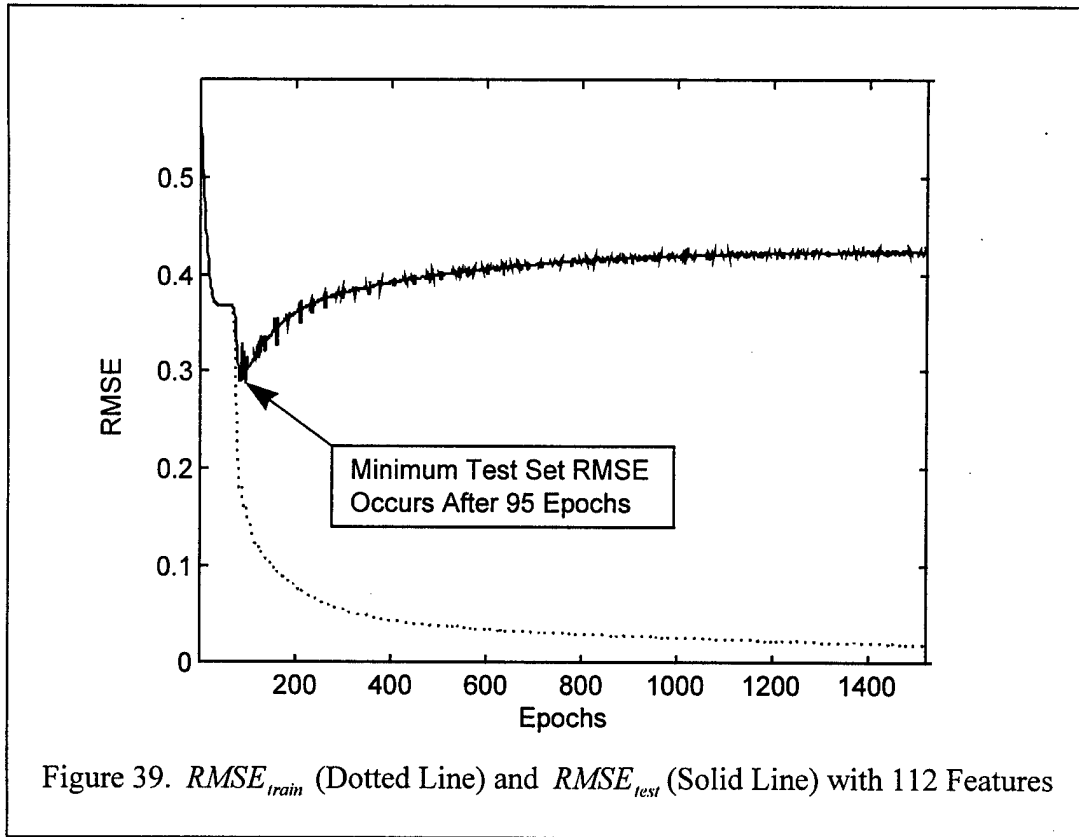
Table 25 provides a summary of $\overline{E}$ required until the minimum $RMSE_{test}$ was attained using 112 features. The design point that produced $\overline{E}$ is shaded in Table 25. The minimum $\overline{E}$ achieved was 84.4 epochs with $m_c = 0.3$ and $J = 5$. The overall $\overline{E}$ until reaching the minimum $RMSE_{test}$ was 94.1. It appears from Table 25 that the design points with $m_c = 0.9$ surprisingly required, on average, the highest $\overline{E}$ in order to attain

Table 24. Average Minimum Test Set RMSE Using 112 Features

		Momentum Constant					
		0.1	0.3	0.5	0.7	0.9	Avg
Number of Hidden/ Context Nodes	1	0.2896	0.2903	0.2941	0.3042	0.3120	0.2981
	3	0.2893	0.2899	0.2896	0.2942	0.3040	0.2934
	5	0.2897	0.2911	0.2907	0.2929	0.2940	0.2916
	7	0.2901	0.2910	0.2903	0.2924	0.2913	0.2910
	9	0.2906	0.2914	0.2902	0.2918	0.2896	0.2907
	Avg	0.2898	0.2907	0.2910	0.2951	0.2982	0.2930

191

Table 25. Average Number of Epochs Using 112 Features

		Momentum Constant					
		0.1	**0.3**	**0.5**	**0.7**	**0.9**	**Avg**
Number of Hidden/ Context Nodes	1	89.3	90.0	89.6	94.4	114.2	95.5
	3	95.3	86.1	88.6	90.7	112.5	94.6
	5	98.2	84.4	86.2	87.4	110.4	93.3
	7	97.5	85.1	86.5	87.2	108.6	93.0
	9	98.8	88.5	85.8	89.8	107.9	94.2
	Avg	95.8	86.8	87.3	89.9	110.7	94.1

the minimum $RMSE_{test}$.

For the training sessions depicted in Figure 39, the set of weights from the 95th epoch was kept. Figure 40 contains plots of the actual output and of the desired output using 112 features using that set of weights at the design point with $m_c = 0.1$ and $J = 3$.

7.4.4.2 Step Two: Perform Signal-to-Noise Ratio (SNR) Screening Method over Experimental Design

The second step of this preliminary investigation was to perform the SNR screening method 250 times following the experimental design. Number of eye blinks was selected as the only salient feature required 248 times (99.2%). From a psychophysiological viewpoint, the number of eye blinks makes sense since number of eye blinks is a good indicator of visual workload and landing an airplane is a visually demanding task [175]. One replication with $m_c = 0.1$ and $J = 1$ selected number of eye blinks and the variance of the log power of the α frequency band at electrode C4 as the parsimonious set of salient features (0.4%). One replication with $m_c = 0.1$ and $J = 5$ selected all 112 features as the parsimonious set of salient features (0.4%). Figure 41 plots the SNR saliency measure of the number of eye blinks and of the variance of the log

192
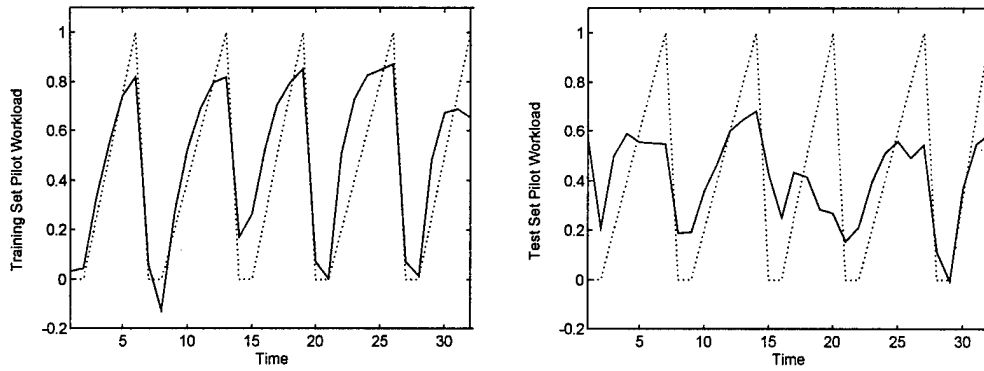
Figure 40. Actual Output (Solid Line) And Desired Output (Dotted Line) With 112
Features

power of the α frequency band at electrode C4 after each feature is removed. The

replication depicted in Figure 41 had $m_c = 0.9$ and $J = 1$. Note from Figure 41 that the

SNR saliency measure of the variance of the log power of the α frequency band at

electrode C4 drops to 0.0 when it is the 111th feature removed. It appears from Figure 41

that the SNR saliency measure of the number of eye blinks is always larger than that of

the variance of the log power of the α frequency band at electrode C4.

Figure 42 plots the SNR saliency measure of the average log power of the Δ

frequency band at electrode P3 and of the average log power of the α_2 frequency band at

electrode FZ. The replication depicted in Figure 42 had $m_c = 0.9$ and $J = 1$. Note from

Figure 42 that average log power of the α_2 frequency band at electrode Fz is the 36th

feature removed. As such, the SNR saliency measure of the average log power of the α_2

frequency band at electrode Fz is 0.0 after it is removed. Also note from Figure 42 that

average log power of the Δ frequency band at electrode P3 is the 106th feature removed.

Figure 43 shows the effect of each feature's removal on $RMSE_{train}$ and $RMSE_{test}$.

The replication depicted in Figure 43 had $m_c = 0.9$ and $J = 1$. The minimum $RMSE_{test}$

193

Figure 41. SNR Saliency Measure of Number of Eye Blinks (Solid Line) and of Variance of Log Power of the α Frequency Band at Electrode C4
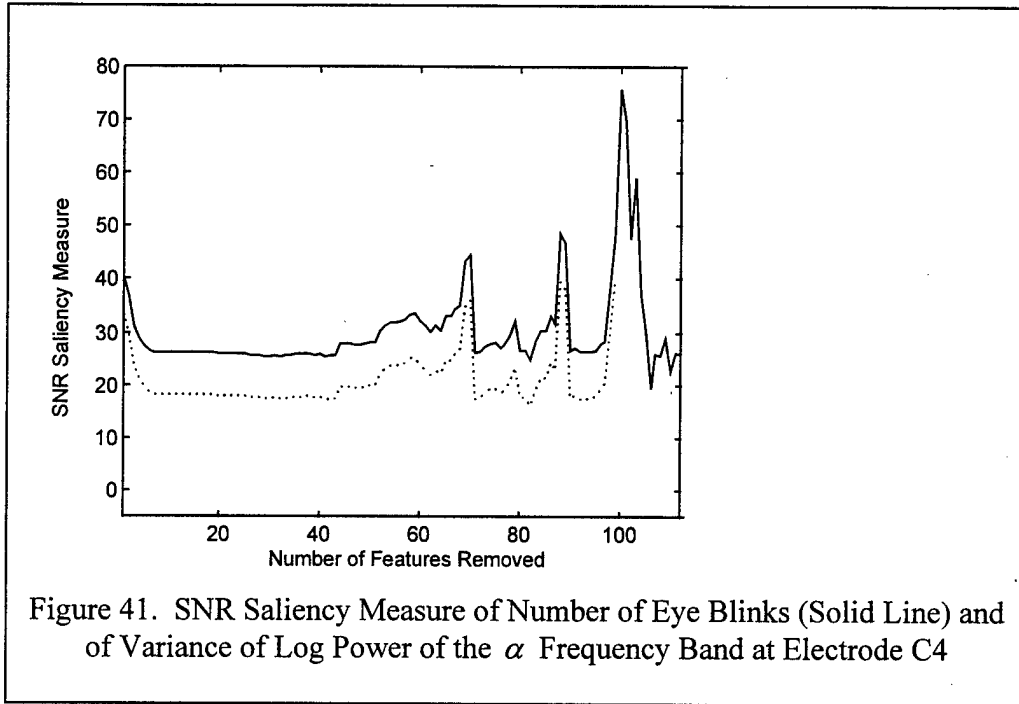
occurred after 111 features were removed and the number of eye blinks was the only feature remaining. This replication selected number of eye blinks as the only salient feature required to estimate a pilot's workload while landing an airplane.

7.4.4.3 Step Three: Train Using Parsimonious Set of Salient Features over Experimental Design

The third and final step was to train 250 Elman RNNs using the parsimonious set of salient features over the experimental design. Figure 44 is a plot of the $RMSE_{train}$ and the $RMSE_{test}$ for one of the training sessions with $m_c = 0.9$ and $J = 7$. For the training session depicted in Figure 44, the minimum $RMSE_{test} = 0.0904$ occurred after $E = 1936$ epochs of training. Training was stopped after $E = 2000$ epochs. After $E = 2000$ epochs, $RMSE_{train} = 0.0759$.

194

Figure 42. SNR Saliency Measure of Average Log Power of Δ Frequency Band at Electrode P3 (Solid Line) and of Average Log Power of α_2 Frequency Band at Electrode Fz (Dotted Line)
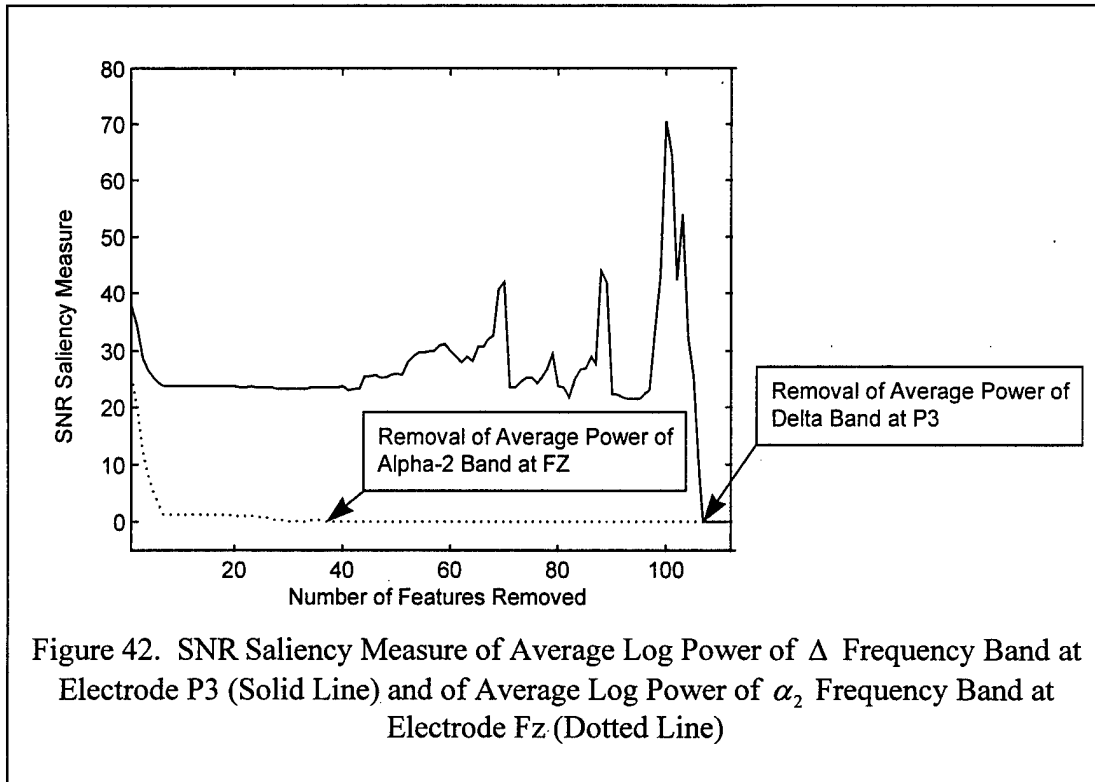
Table 26 provides a summary of the minimum $\overline{RMSE}_{test}$ using number of eye blinks. It appears from Table 26 that m_c and J have a small but noticeable effect on $\overline{RMSE}_{test}$. The design point that produced the best $\overline{RMSE}_{test}$ is shaded in Table 26. The best $\overline{RMSE}_{test} = 0.0864$ was achieved with $m_c = 0.9$ and $J = 7$. This provided a 70% reduction from the best $\overline{RMSE}_{test} = 0.2893$ attained using all 112 features with $m_c = 0.1$ and $J = 3$. The overall $\overline{RMSE}_{test}$ using number of eye blinks was 0.0969. The overall average test set RMSE using all 112 features was 0.2930. The SNR screening method provided, on average, a 67% reduction in $RMSE_{test}$. By comparing Table 26 with Table 24, it is clear that results from the SNR screening method produce a lower $\overline{RMSE}_{test}$ regardless of m_c and J.

Figure 43. $RMSE_{train}$ (Dotted Line) and $RMSE_{test}$ (Solid Line) after Each
Feature is Removed
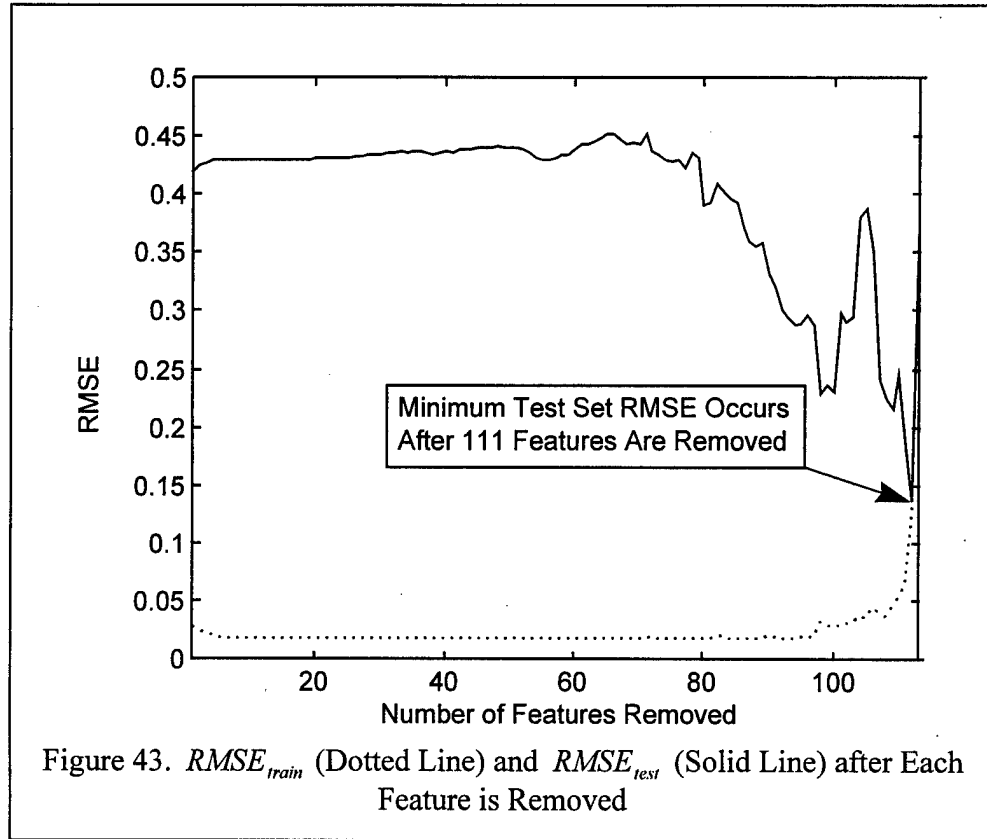
Table 27 provides a summary of $\overline{E}$ required until the minimum $RMSE_{test}$ is attained using number of eye blinks. The design point that produced the best $\overline{E}$ is shaded in Table 27. The minimum $\overline{E}$ required using number of eye blinks was 1498.8 epochs with $m_c = 0.9$ and $J = 3$. The minimum $\overline{E}$ required using all 112 features was 84.4 epochs with $m_c = 0.3$ and $J = 5$. The minimum $\overline{E}$ required using number of eye blinks is larger than that using all 112 features by a factor of 17.75. The overall $\overline{E}$ using number of eye blinks was 1903.4. The overall $\overline{E}$ using all 112 features was 94.1. The overall $\overline{E}$ required using number of eye blinks is larger than that using all 112 features by a factor of 20.22. By comparing Table 27 with Table 25, it is clear that it takes a larger E, on average, to train an Elman RNN using number of eye blinks regardless of

196

Figure 44. $RMSE_{train}$ (Dotted Line) and $RMSE_{test}$ (Solid Line) with Number Of Eye Blinks
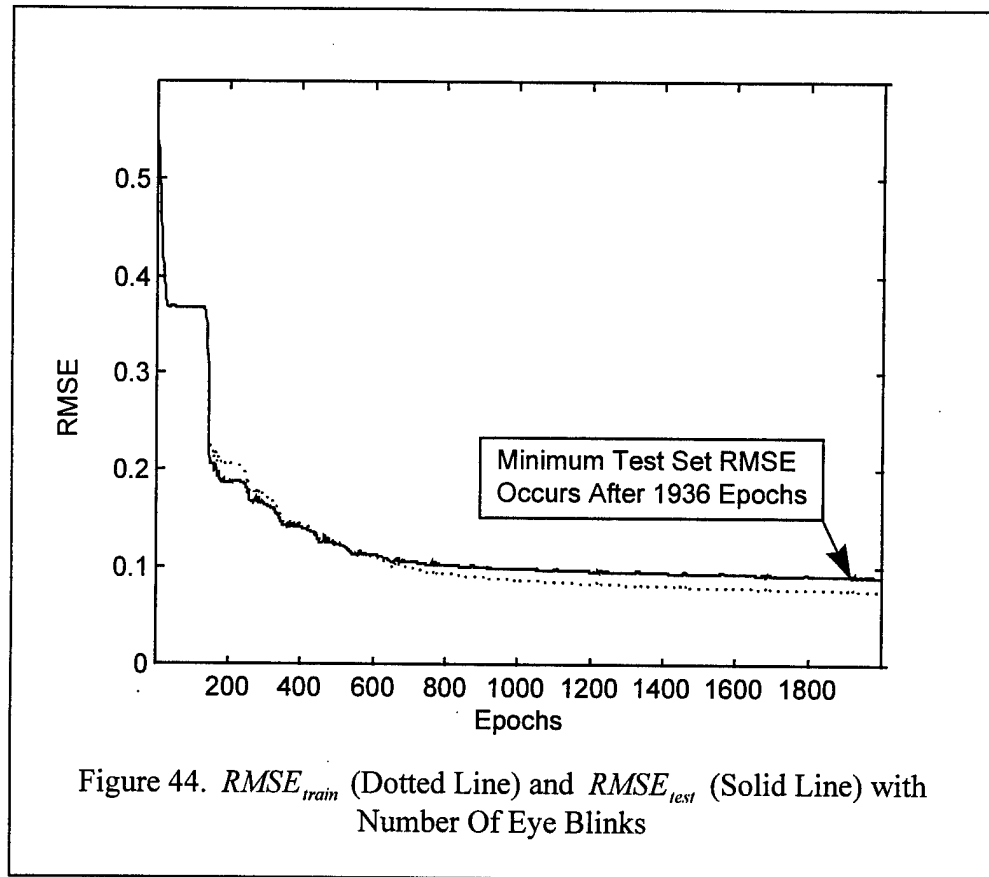
m_c and J. It may take longer to train with number of eye blinks as the only input feature, but the added epochs provide a much lower $RMSE_{test}$.

Figure 45 is a plot of the actual output and of the desired output for both the training set and the test set. Results depicted in Figure 45 are from an Elman RNN for the training session depicted in Figure 44 where the set of weights from $E = 1936$ are kept, $m_c = 0.9$ and $J = 7$. By comparing Figure 45 with Figure 40, it is clear that using only the number of eye blinks provided better results to estimating a pilot's workload while landing an airplane.

197

Table 26. Minimum $\overline{RMSE}_{test}$ Using Number Of Eye Blinks

		Momentum Constant					
		0.1	0.3	0.5	0.7	0.9	Avg
Number of Hidden/ Context Nodes	1	0.1192	0.1155	0.1122	0.1088	0.1041	0.1120
	3	0.0954	0.0944	0.0941	0.0899	0.0906	0.0929
	5	0.0975	0.0897	0.0891	0.0915	0.0871	0.0910
	7	0.1010	0.0958	0.0882	0.0900	0.0864	0.0923
	9	0.1038	0.0978	0.0950	0.0944	0.0900	0.0962
	Avg	0.1034	0.0986	0.0957	0.0949	0.0916	0.0969

7.4.5 Conclusions

In conclusion, this application of the SNR screening method was the first successful USAF effort to estimate a pilot's workload while landing an airplane using EEG and psychophysiological features via an Elman RNN. This was also the first successful application of the SNR screening method to an Elman RNN. The overall minimum $\overline{RMSE}_{test}$ using all 112 available features was 0.2930. Conclusions drawn from applying the SNR screening method to the Elman RNN consistently stated that number of eye blinks was the only salient feature required. The overall $\overline{RMSE}_{test}$ using number of eye blinks was 0.0969.

Table 27. $\overline{E}$ Using Number of Eye Blinks

		Momentum Constant					
		0.1	0.3	0.5	0.7	0.9	Avg
Number of Hidden/ Context Nodes	1	1991.3	1990.8	1992.4	1998.3	1997.8	1994.1
	3	1943.1	1922.2	1870.7	1691.5	1489.8	1783.5
	5	1993.4	1980.4	1887.2	1873.1	1623.7	1871.6
	7	1721.3	1986.4	1990.0	1988.5	1890.7	1915.4
	9	1854.7	1949.1	1986.4	1984.0	1988.9	1952.6
	Avg	1900.8	1965.8	1945.3	1907.1	1798.2	1903.4

198
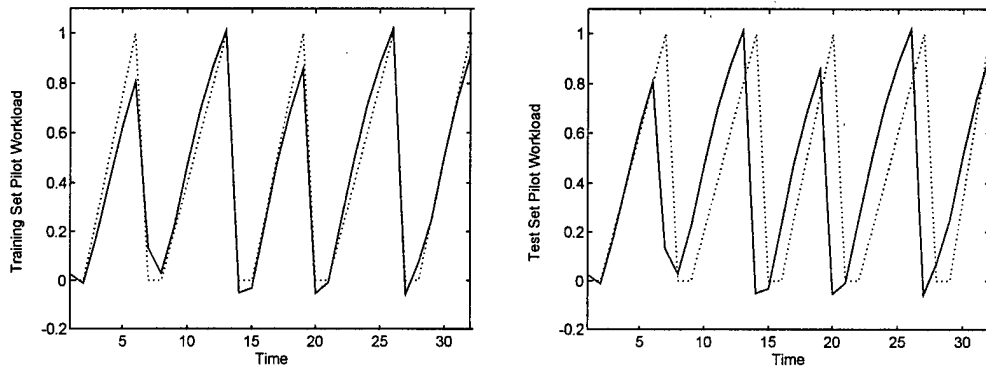
Figure 45. Actual Output (Solid Line) And Desired Output (Dotted Line) With Number of Eye Blinks

7.5 Conclusions

The SNR screening method, as demonstrated in the pilot workload classification and estimation problems, identified and removed non-salient features. The SNR screening method shows promise and offers significant advantages over previous screening methods. A significant advantage of the SNR saliency measure is that the saliency of each feature is compared to that of a known nonsalient noise feature. A significant advantage of the SNR screening method over the Belue-Bauer screening method and the Steppe-Bauer screening method is that the screening may be completed in only one training run. The SNR screening method is able to select a parsimonious set of salient features while maintaining, and in some cases decreasing, $\overline{RMSE}_{test}$. Finally, the SNR screening method shows some potential for identifying redundant features.

8   Spatial-Temporal Feature Screening Method that Utilizes a Partial Derivative-Based Spatial-Temporal Saliency Measure

8.1   Introduction

Unfortunately, the SNR saliency measure as used in an Elman RNN in the previous chapter does not explicitly account for the temporal saliency of each feature. In response to this, this research derived and developed a partial derivative-based spatial-temporal saliency measure for use in Elman RNNs. This partial derivative-based saliency measure provides the spatial-temporal saliency of each feature by unfolding the layers of an Elman RNN through time. This chapter discusses the development of a partial derivative-based spatial-temporal saliency measure to be used in Elman RNNs.

In order to use the partial derivative-based spatial-temporal saliency measure for feature selection, a spatial-temporal feature screening method was developed. As with other screening methods developed in this dissertation, a noise feature is injected. In the spatial-temporal feature screening method, features are screened out by comparing the area under the spatial-temporal curves. The applicability of the new methodology is exhibited by applying it to classifying pilot workload. The features include those derived from peripheral psychophysiological quantities. The new spatial-temporal feature screening method and the new partial derivative-based spatial-temporal feature saliency measure is useful in determining the relevance of peripheral psychophysiological features for classifying pilot workload over time.

This chapter is organized as follows. First, derivations for the partial derivative-based saliency measure are provided for a simple $1+1/1/1$ Elman RNN, a $1+J/J/1$

Elman RNN, and a general $I + J / J / K$ Elman RNN. Next, the spatial-temporal feature screening method is described. Finally, the applicability of the spatial-temporal feature screening method is shown for classifying pilot workload using peripheral psychophysiological features. To show the advantages of the using Elman RNNs for classifying pilot workload, feature screening is also performed using the SNR screening method for a feedforward MLP ANN and a TDNN. Then the results using an Elman RNN are compared to that using a feedforward MLP ANN and a TDNN.

8.2 Partial Derivative-Based Spatial-Temporal Saliency Measure

This section derives the partial derivative-based spatial-temporal saliency measure denoted as Γ for varying complexity of Elman RNN architectures. First, the derivations for Γ are provided for a very simple $1 + 1 / 1 / 1$ Elman RNN. Then, derivations are provided for a $1 + J / J / 1$ Elman RNN. Finally, derivations are provided for a general $I + J / J / K$ Elman RNN. This bottom-up approach to deriving Γ is provided as a convenience to the reader.

8.2.1 Derivations for a $1 + 1 / 1 / 1$ Elman Recurrent Neural Network (RNN)

Figure 46 shows a $1 + 1 / 1 / 1$ Elman RNN which is simplest Elman RNN architecture. Let $f(a)$ denote an activation function and $\dot{f}$ denote $\dfrac{\partial f}{\partial a}$. Given a function $x_1(t)$ for $t \in \{1, 2, 3, \ldots\}$, then define the function $a_1^1(t, \mathbf{W})$ as:

$$a_1^1(t, \mathbf{W}) = w_{0,1}^1 + w_{1,1}^1 \cdot x_1(t) + w_{2,1}^1 \cdot y_1(t-1) \tag{109}$$

where $a_1^1(t, \mathbf{W})$ is the activation argument for hidden node $j = 1$ and for $layer = 1$. The

201

subscript of $a_1^1(t, \mathbf{W})$ denotes hidden node $j = 1$ and the superscript denotes *layer* = 1. Then define:

$$y_1(t, \mathbf{W}) = f_1^1(a_1^1(t, \mathbf{W})) \tag{110}$$

where the subscript of f_1^1 denotes hidden node $j = 1$ and the superscript denotes *layer* = 1. Now, define the function $a_1^2(t, \mathbf{W})$ for $t \in \{1,2,3,\ldots\}$ as:

$$a_1^2(t, \mathbf{W}) = w_{0,1}^2 + w_{1,1}^2 \cdot y_1(t, \mathbf{W}) \tag{111}$$

where the subscript of $a_1^2(t, \mathbf{W})$ denotes output node $k = 1$ and the superscript denotes *layer* = 2. Then define:

$$z_1(t, \mathbf{W}) = f_1^2(a_1^2(t, \mathbf{W})) \tag{112}$$

where the subscript of f_1^2 denotes output node $k = 1$ and the superscript denotes *layer* = 2. Note that $z_1(t, \mathbf{W})$ depends upon the parameters $w_{0,1}^1$, $w_{1,1}^1$, $w_{2,1}^1$, $w_{0,1}^2$, and

Figure 46. 1 + 1 / 1 / 1 Elman RNN

202
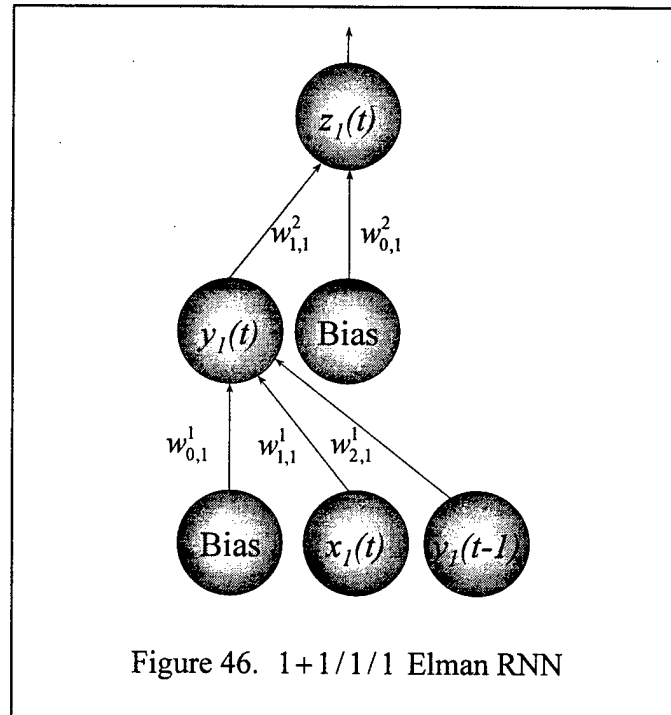
$w_{1,1}^2$. Hence, $z_1(t, \mathbf{W})$ is a function of $\mathbf{W}$, the weight matrix. The function z_1 clearly depends on x_1 and so, $z_1(t, \mathbf{W})$ is actually:

$$z_1(t, \mathbf{W}) = z_1(x_1(0), x_1(1), \ldots, x_1(t), \mathbf{W}) \tag{113}$$

For t fixed let $\widetilde{z}_1(\xi_0, \xi_1, \xi_2 \ldots, \xi_t, W)$ denote the function so that:

$$z_1(t, \mathbf{W}) = \widetilde{z}_1(x_1(0), x_1(1), x_1(2), \ldots, x_1(t), \mathbf{W}) \tag{114}$$

Therefore, the instantaneous rate of change of $z_1(t, \mathbf{W})$ with respect to $x_1(t)$ is written as:

$$\frac{\partial \widetilde{z}_1}{\partial \xi_t}(x_1(0), x_1(1), x_1(2), \ldots, x_1(t), \mathbf{W}) \tag{115}$$

Equation 115 will be written in abbreviated form as:

$$\frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t)} \tag{116}$$

Applying the chain rule, Equation 116 becomes:

$$\frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t)} = \dot{f}_1^2\left(w_{0,1}^2 + w_{1,1}^2 \cdot y_1(t)\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(w_{0,1}^1 + w_{1,1}^1 \cdot x_1(t) + w_{2,1}^1 \cdot y_1(t-1)\right) \cdot w_{1,1}^1 \tag{117}$$

Equation 117 can be rewritten as:

$$\frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t)} = \dot{f}_1^2\left(a_1^2(t, \mathbf{W})\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(a_1^1(t, \mathbf{W})\right) \cdot w_{1,1}^1 \tag{118}$$

For the $1+1/1/1$ Elman RNN as depicted in Figure 46, the partial derivative-based saliency measure is calculated for feature x_1, using the training set exemplars following Equation 66 as:

$$\Gamma_{x_1} = \frac{1}{\mathrm{T}} \cdot \sum_{t=1}^{\mathrm{T}} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t)} \right| \tag{119}$$

where Γ_{x_1} is the partial derivative-based saliency measure for feature x_1, T is the

number of time steps, $z_1(t, \mathbf{W})$ is the activation of output z_1 at time t with the trained weight matrix $\mathbf{W}$, and $x_1(t)$ is the input for feature x_1 at time t. More specifically:

$$\Gamma_{x_1} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2(a_1^2(t, \mathbf{W})) \cdot w_{1,1}^2 \cdot \dot{f}_1^1(a_1^1(t, \mathbf{W})) \cdot w_{1,1}^1 \right| \tag{120}$$

where $\dot{f}_1^2(t, \mathbf{W})$ is the partial derivative of $z_1(t, \mathbf{W})$, $w_{1,1}^2$ is the second layer weight connecting hidden node $y_1(t)$ to the output node $z_1(t)$, $\dot{f}_1^1(t, \mathbf{W})$ is the partial derivative of $y_1(t, \mathbf{W})$, $y_1(t, \mathbf{W})$ is the activation of hidden node y_1 at time t with the trained weight matrix $\mathbf{W}$, and $w_{1,1}^1$ is the first layer weight connecting input node $x_1(t)$ to hidden node $y_1(t)$. The partial derivatives of various transfer functions are listed in Table 1. Similarly to Equation 119, the partial derivative-based saliency measure can be computed for the context node y_1 as:

$$\Gamma_{y_1} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial y_1(t-1, \mathbf{W})} \right| \tag{121}$$

where Γ_{y_1} is the partial derivative-based saliency measure for context node y_1. More specifically:

$$\Gamma_{y_1} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2(a_1^2(t, \mathbf{W})) \cdot w_{1,1}^2 \cdot \dot{f}_1^1(a_1^1(t, \mathbf{W})) \cdot w_{2,1}^1 \right| \tag{122}$$

where $w_{2,1}^1$ is the first layer weight connecting context node $y_1(t-1)$ to hidden node $y_1(t)$. Note the following relationship:

$$\frac{\Gamma_{x_1}}{\Gamma_{y_1}} = \frac{\frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2(a_1^2(t, \mathbf{W})) \cdot w_{1,1}^2 \cdot \dot{f}_1^1(a_1^1(t, \mathbf{W})) \cdot w_{1,1}^1 \right|}{\frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2(a_1^2(t, \mathbf{W})) \cdot w_{1,1}^2 \cdot \dot{f}_1^1(a_1^1(t, \mathbf{W})) \cdot w_{2,1}^1 \right|} = \frac{|w_{1,1}^1|}{|w_{2,1}^1|} \tag{123}$$

204

8.2.2 One Time Lag of a $1+1/1/1$ Elman Recurrent Neural Network (RNN)

An Elman RNN can be viewed as a feedforward MLP ANN which has been folded back on itself in time. An $1+1/1/1$ Elman RNN can be unfolded one layer as shown in Figure 47. Unfolding a layer of an Elman RNN allows us to visualize the input and hidden layers that effect the output $z_1(t)$. The partial derivative-based saliency measure can be calculated for the first unfolded layer by extending the equations in Section 8.2.1. For the unfolded layer 1 in Figure 47, the partial derivative-based saliency measure for x_1 is calculated as:

$$\Gamma_{x_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t-1)} \right| \tag{124}$$

Figure 47. $1+1/1/1$ Elman RNN Unfolded One Layer

205
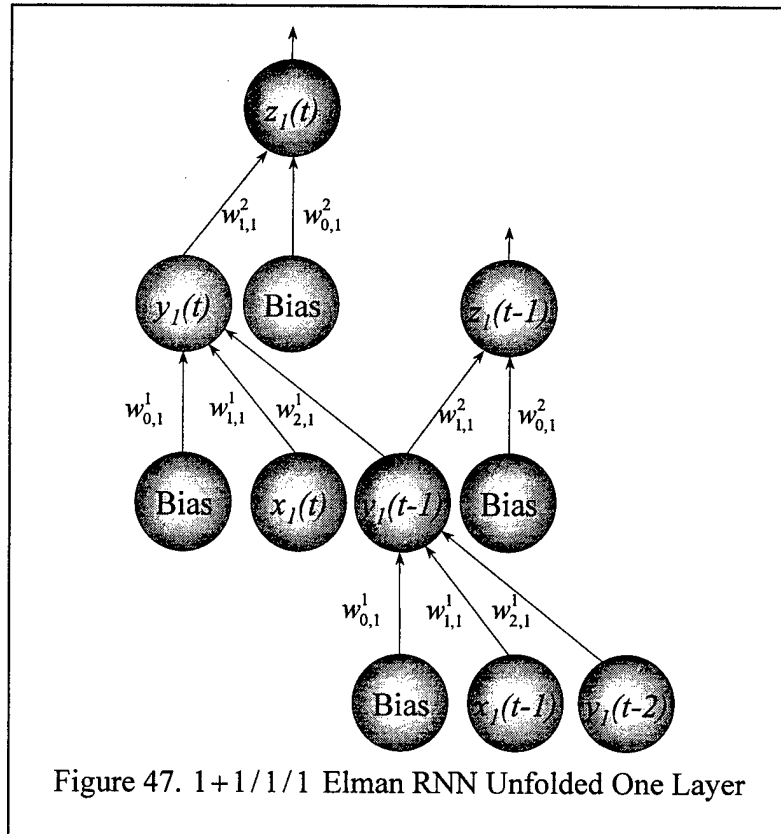
where $\Gamma_{x_1}^1$ is the partial derivative-based saliency measure for feature x_1 on unfolded layer 1.

The superscript of Γ denotes the unfolded lag $\ell = 1,2,\ldots L$. If $\ell = 0$, then no superscript is used on Γ to clarify the use of the classical version of the partial derivative-based saliency measure as described in Section 3.4.2. The subscript of Γ denotes the type of node. In the case of a $1+1/1/1$ Elman RNN, the type of node is either input node x_1 or context node y_1. Equation 124 can be written more specifically as:

$$\Gamma_{x_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2 \left(a_1^2(t,\mathbf{W}) \right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1 \left(a_1^1(t,\mathbf{W}) \right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1 \left(a_1^1(t-1,\mathbf{W}) \right) \cdot w_{1,1}^1 \right| \quad (125)$$

Note that $w_{1,1}^1$ connecting input node $x_1(t-1)$ to hidden node $y_1(t-1)$ is the same as $w_{1,1}^1$ connecting input node $x_1(t)$ to hidden node $y_1(t)$. The weights in an Elman RNN are not dynamic.

Similarly to Equation 124, the partial derivative-based saliency measure can be computed for the context node y_1 on unfolded layer 1 as:

$$\Gamma_{y_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial y_1(t-2,\mathbf{W})} \right| \quad (126)$$

where $\Gamma_{y_1}^1$ is the partial derivative-based saliency measure for context node y_1 on unfolded layer 1. More specifically:

$$\Gamma_{y_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2 \left(a_1^2(t,\mathbf{W}) \right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1 \left(a_1^1(t,\mathbf{W}) \right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1 \left(a_1^1(t-1,\mathbf{W}) \right) \cdot w_{2,1}^1 \right| \quad (127)$$

Note the following relationship:

$$\frac{\Gamma_{x_1}^1}{\Gamma_{y_1}^1} = \frac{\frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2\left(a_1^2(t, \mathbf{W})\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(a_1^1(t, \mathbf{W})\right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-1, \mathbf{W})\right) \cdot w_{1,1}^1 \right|}{\frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2\left(a_1^2(t, \mathbf{W})\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(a_1^1(t, \mathbf{W})\right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-1, \mathbf{W})\right) \cdot w_{2,1}^1 \right|} = \frac{\left| w_{1,1}^1 \right|}{\left| w_{2,1}^1 \right|} \quad (128)$$

The results in Equation 128 are the same as that in Equation 123.

8.2.3   Two Time Lags of a 1+1/1/1 Elman Recurrent Neural Network (RNN)

The unfolding of an Elman RNN can be continued further. Figure 48 shows a 1+1/1/1 Elman RNN unfolded two layers. This unfolding of an Elman RNN shows the
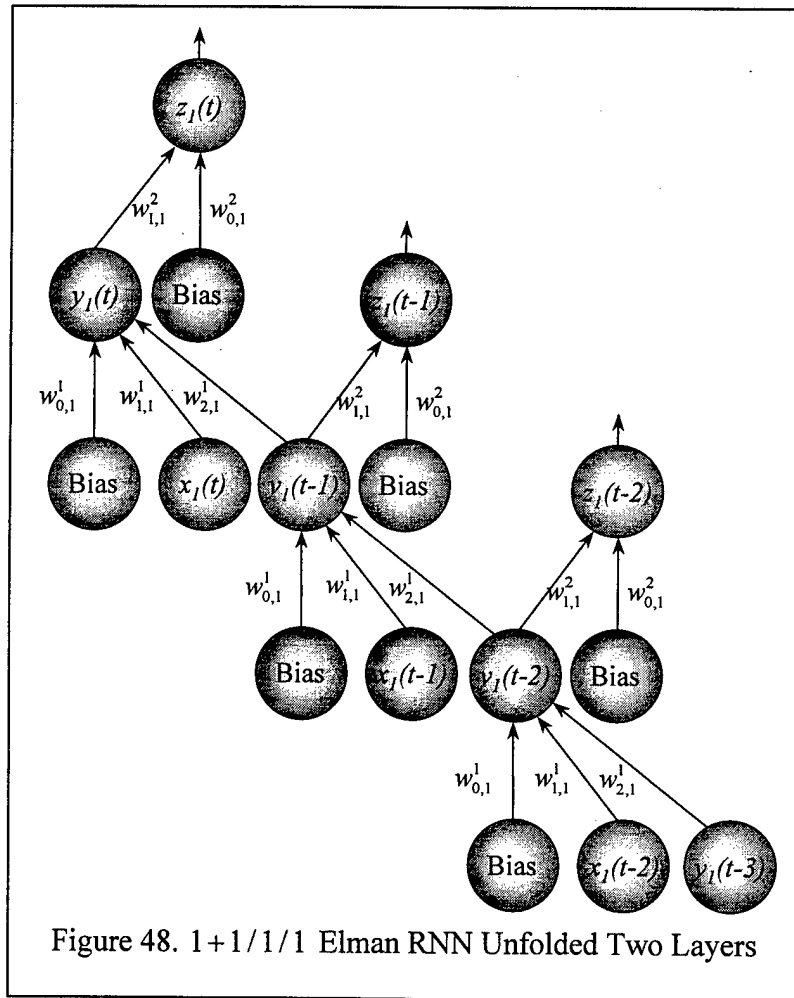
Figure 48. 1+1/1/1 Elman RNN Unfolded Two Layers

effect of the temporal feedback in a spatial representation. The partial derivative-based saliency measure can be calculated for the second unfolded layer by extending the equations in Section 8.2.1 and Section 8.2.2. For the unfolded layer 2 in Figure 48, the partial derivative-based saliency measure for x_1 is calculated as:

$$\Gamma_{x_1}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t-2)} \right| \tag{129}$$

where $\Gamma_{x_1}^2$ is the partial derivative-based saliency measure for feature x_1 on unfolded layer 2. More specifically:

$$\Gamma_{x_1}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \begin{array}{l} \dot{f}_1^2\big(a_1^2(t, \mathbf{W})\big) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\big(a_1^1(t, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \\ \dot{f}_1^1\big(a_1^1(t-1, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\big(a_1^1(t-2, \mathbf{W})\big) \cdot w_{1,1}^1 \end{array} \right| \tag{130}$$

Similarly to Equation 129, the partial derivative-based saliency measure can be computed for the context node y_1 on unfolded layer 2 as:

$$\Gamma_{y_1}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial y_1(t-3, \mathbf{W})} \right| \tag{131}$$

where $\Gamma_{y_1}^2$ is the partial derivative-based saliency measure for context node y_1 on unfolded layer 2. More specifically:

$$\Gamma_{y_2}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \begin{array}{l} \dot{f}_1^2\big(a_1^2(t, \mathbf{W})\big) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\big(a_1^1(t, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \\ \dot{f}_1^1\big(a_1^1(t-1, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\big(a_1^1(t-2, \mathbf{W})\big) \cdot w_{2,1}^1 \end{array} \right| \tag{132}$$

Note the following relationship:

$$\frac{\Gamma_{x_1}^2}{\Gamma_{y_1}^2} = \frac{\dfrac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \begin{array}{l} \dot{f}_1^2\big(a_1^2(t, \mathbf{W})\big) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\big(a_1^1(t, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \\ \dot{f}_1^1\big(a_1^1(t-1, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\big(a_1^1(t-2, \mathbf{W})\big) \cdot w_{1,1}^1 \end{array} \right|}{\dfrac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \begin{array}{l} \dot{f}_1^2\big(a_1^2(t, \mathbf{W})\big) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\big(a_1^1(t, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \\ \dot{f}_1^1\big(a_1^1(t-1, \mathbf{W})\big) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\big(a_1^1(t-2, \mathbf{W})\big) \cdot w_{2,1}^1 \end{array} \right|} = \frac{\left| w_{1,1}^1 \right|}{\left| w_{2,1}^1 \right|} \tag{133}$$

As with Equation 123 and Equation 128, Equation 133 has the same results.

8.2.4 N Time Lags of a $1+1/1/1$ Elman Recurrent Neural Network (RNN)

Continuing on with the unfolding, a $1+1/1/1$ Elman RNN can be unfolded N layers as shown in Figure 49. The partial derivative-based saliency measure can be calculated for the Nth unfolded layer by extending the equations in Sections 8.2.1 through 8.2.3. For the unfolded layer N in Figure 49, the partial derivative-based saliency measure for x_1 is calculated as:

$$\Gamma_{x_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial x_1(t-N)} \right| \tag{134}$$

where $\Gamma_{x_1}^N$ is the partial derivative-based saliency measure for context node x_1 on unfolded layer N. More specifically:

$$\Gamma_{x_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T} \left| \begin{array}{l} \dot{f}_1^2\left(a_1^2(t,\mathbf{W})\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(a_1^1(t,\mathbf{W})\right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-1,\mathbf{W})\right) \cdot \\ w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-2,\mathbf{W})\right) \cdot \ldots \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-N,\mathbf{W})\right) \cdot w_{1,1}^1 \end{array} \right| \tag{135}$$

Similarly to Equation 134, the partial derivative-based saliency measure can be computed for the context node y_1 on unfolded layer N as:

$$\Gamma_{y_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial y_1(t-N-1,\mathbf{W})} \right| \tag{136}$$

where $\Gamma_{y_1}^N$ is the partial derivative-based saliency measure for context node y_1 on unfolded layer N. More specifically:

$$\Gamma_{y_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \begin{array}{l} \dot{f}_1^2\left(a_1^2(t,\mathbf{W})\right) \cdot w_{1,1}^2 \cdot \dot{f}_1^1\left(a_1^1(t,\mathbf{W})\right) \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-1,\mathbf{W})\right) \cdot \\ w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-2,\mathbf{W})\right) \cdot \ldots \cdot w_{2,1}^1 \cdot \dot{f}_1^1\left(a_1^1(t-N,\mathbf{W})\right) \cdot w_{2,1}^1 \end{array} \right| \tag{137}$$

Figure 49. $1+1/1/1$ Elman RNN Unfolded N Layers

Note the following relationship:

$$\frac{\Gamma_{x_1}^N}{\Gamma_{y_1}^N} = \frac{\dfrac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \begin{array}{l} \dot{f}_1^2\!\left(a_1^2(t,\mathbf{W})\right)\cdot w_{1,1}^2 \cdot \dot{f}_1^1\!\left(a_1^1(t,\mathbf{W})\right)\cdot w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-1,\mathbf{W})\right)\cdot \\ w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-2,\mathbf{W})\right)\cdot \ldots \cdot w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-N,\mathbf{W})\right)\cdot w_{1,1}^1 \end{array} \right|}{\dfrac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \begin{array}{l} \dot{f}_1^2\!\left(a_1^2(t,\mathbf{W})\right)\cdot w_{1,1}^2 \cdot \dot{f}_1^1\!\left(a_1^1(t,\mathbf{W})\right)\cdot w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-1,\mathbf{W})\right)\cdot \\ w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-2,\mathbf{W})\right)\cdot \ldots \cdot w_{2,1}^1 \cdot \dot{f}_1^1\!\left(a_1^1(t-N,\mathbf{W})\right)\cdot w_{2,1}^1 \end{array} \right|} = \frac{\left| w_{1,1}^1 \right|}{\left| w_{2,1}^1 \right|} \quad (138)$$

The result in Equation 138 is the same as that in Equations 123, 128, and 133. In fact the resulting ratio holds for all unfolded layers of an Elman RNN whenever $J = 1$.

8.2.5 Derivations for a $1 + J / J / 1$ Elman Recurrent Neural Network (RNN)

The complexity of an Elman RNN architecture can be increased from a $1 + 1 / 1 / 1$ Elman RNN to a $1 + J / J / 1$ Elman RNN by allowing for J hidden nodes which are feedback onto the input layer as J context nodes. Figure 50 shows a $1 + J / J / 1$ Elman RNN. The partial derivative-based saliency measure for a $1 + J / J / 1$ Elman RNN is calculated for feature x_1, using the training set exemplars following Equation 119 as:

$$\Gamma_{x_1} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t)} \right| \tag{139}$$

More specifically:

$$\Gamma_{x_1} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2\left(a_1^2(t, \mathbf{W})\right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1\left(a_j^1(t, \mathbf{W})\right) \cdot w_{1,j}^1 \right| \tag{140}$$

Figure 50. $1 + J / J / 1$ Elman RNN
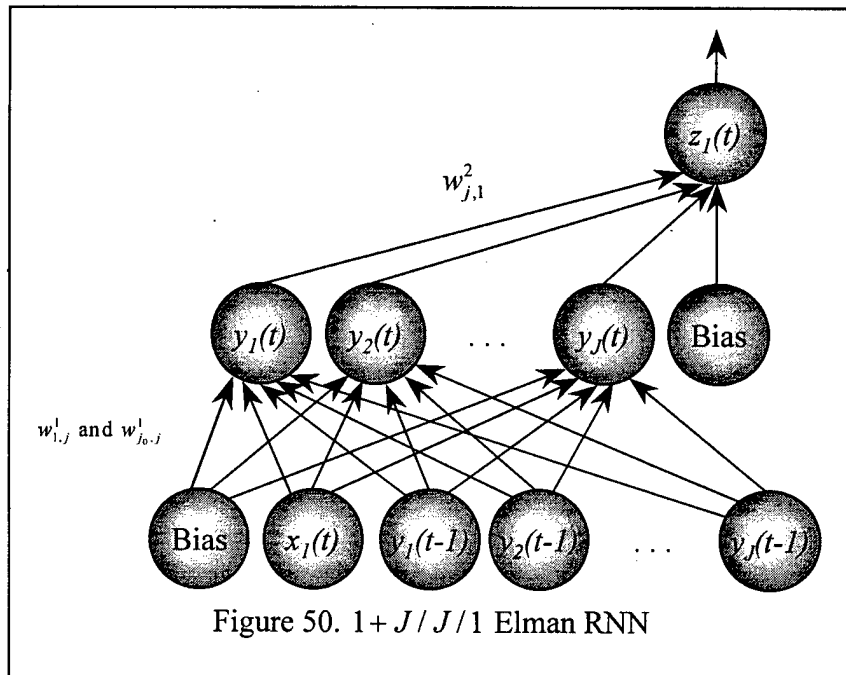
The difference between Equation 120 and Equation 140 is the summation over the hidden nodes. Similarly to Equation 139, the partial derivative-based saliency measure can be computed for context node y_{j_0} for $j_0 = 1, 2, \ldots J$ as:

$$\Gamma_{y_{j_0}} = \frac{1}{\mathrm{T}} \cdot \sum_{t=1}^{\mathrm{T}} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial y_{j_0}(t-1, \mathbf{W})} \right| \tag{141}$$

where $\Gamma_{y_{j_0}}$ is the partial derivative-based saliency measure for context node y_{j_0} for $j_0 = 1, 2, \ldots J$. More specifically:

$$\Gamma_{y_{j_0}} = \frac{1}{\mathrm{T}} \cdot \sum_{t=1}^{\mathrm{T}} \left| \dot{f}_1^2 \left(a_1^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot w_{j_0,j}^1 \right| \tag{142}$$

The difference between Equation 142 and Equation 122 is the summation over the hidden nodes. With the summation found in Equation 140 and Equation 142, however, the ratio found in Equation 123 no longer holds. The summation difference between a $1 + J / J / 1$ Elman RNN and a $1 + 1 / 1 / 1$ Elman RNN will hold for every unfolded layer as described in Sections 8.2.6 through 0. In addition, the ratio of saliency measures will no longer hold true for all unfolded layers in a $1 + J / J / 1$ Elman RNN as described in Sections 8.2.6 through 0.

8.2.6   One Time Lag of a $1 + J / J / 1$ Elman Recurrent Neural Network (RNN)

Just as an $1 + 1 / 1 / 1$ Elman RNN can be unfolded one layer, so can a $1 + J / J / 1$ Elman RNN as shown in Figure 51. The partial derivative-based saliency measure can be calculated for the first unfolded layer by extending the equations in Section 8.2.5. For the unfolded layer 1 in Figure 51, the partial derivative-based saliency measure for x_1 is calculated as:

212

$$\Gamma_{x_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial x_1(t-1)} \right| \qquad (143)$$

More specifically:

$$\Gamma_{x_1}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2\left(a_1^2(t,\mathbf{W})\right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1\left(a_j^1(t,\mathbf{W})\right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1\left(a_{j_0}^1(t-1,\mathbf{W})\right) \cdot w_{1,j_0}^1 \right| \quad (144)$$

Similarly to Equation 143, the partial derivative-based saliency measure can be computed

for context node y_{j_1} for $j_1 = 1, 2, \ldots J$ as:

$$\Gamma_{y_{j_1}}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial y_{j_1}(t-2,\mathbf{W})} \right| \qquad (145)$$

More specifically:

Figure 51. $1 + J / J / 1$ Elman RNN Unfolded One Layer

213
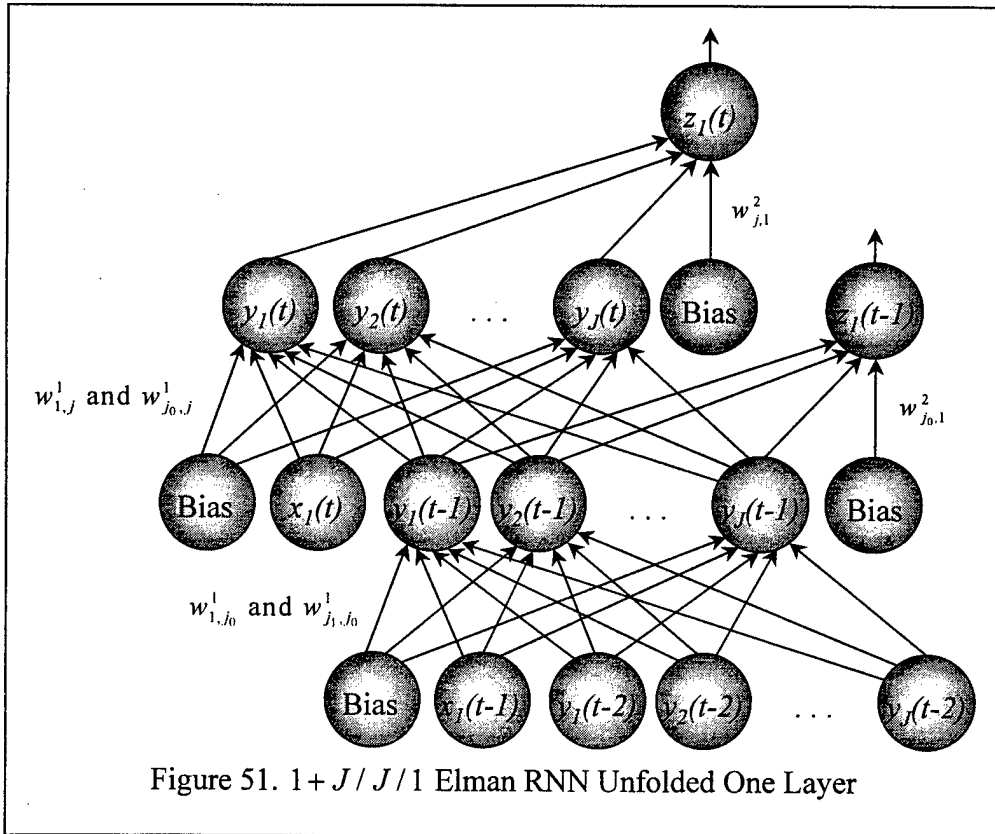
$$\Gamma^1_{y_{j_1}} = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \dot{f}_1^2 \left(a_1^2(t,\mathbf{W})\right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t,\mathbf{W})\right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1,\mathbf{W})\right) \cdot w_{j_1,j_0}^1 \right| \quad (146)$$

8.2.7 Two Time Lags of a $1+J/J/1$ Elman Recurrent Neural Network (RNN)

The unfolding of an Elman RNN can be continued further. Figure 52 shows a $1+J/J/1$ Elman RNN unfolded two layers. The partial derivative-based saliency measure can be calculated for the second unfolded layer by extending the equations in Section 8.2.5 and Section 8.2.6. For the unfolded layer 2 in Figure 52, the partial derivative-based saliency measure for x_1 is calculated as:

$$\Gamma^2_{x_1} = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial x_1(t-2)} \right| \quad (147)$$

More specifically:

$$\Gamma^2_{x_1} = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \dot{f}_1^2 \left(a_1^2(t,\mathbf{W})\right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t,\mathbf{W})\right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1,\mathbf{W})\right) \cdot \right.$$
$$\left. \sum_{j_1=1}^{J} w_{j_1,j_0}^1 \cdot \dot{f}_{j_1}^1 \left(a_{j_1}^1(t-2,\mathbf{W})\right) \cdot w_{1,j_1}^1 \right| \quad (148)$$

Similarly to Equation 147, the partial derivative-based saliency measure can be computed for context node y_{j_2} for $j_2 = 1,2,\ldots J$ as:

$$\Gamma^2_{y_{j_2}} = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \frac{\partial z_1(t,\mathbf{W})}{\partial y_{j_2}(t-3,\mathbf{W})} \right| \quad (149)$$

More specifically:

Figure 52. $1 + J / J / 1$ Elman RNN Unfolded Two Layers
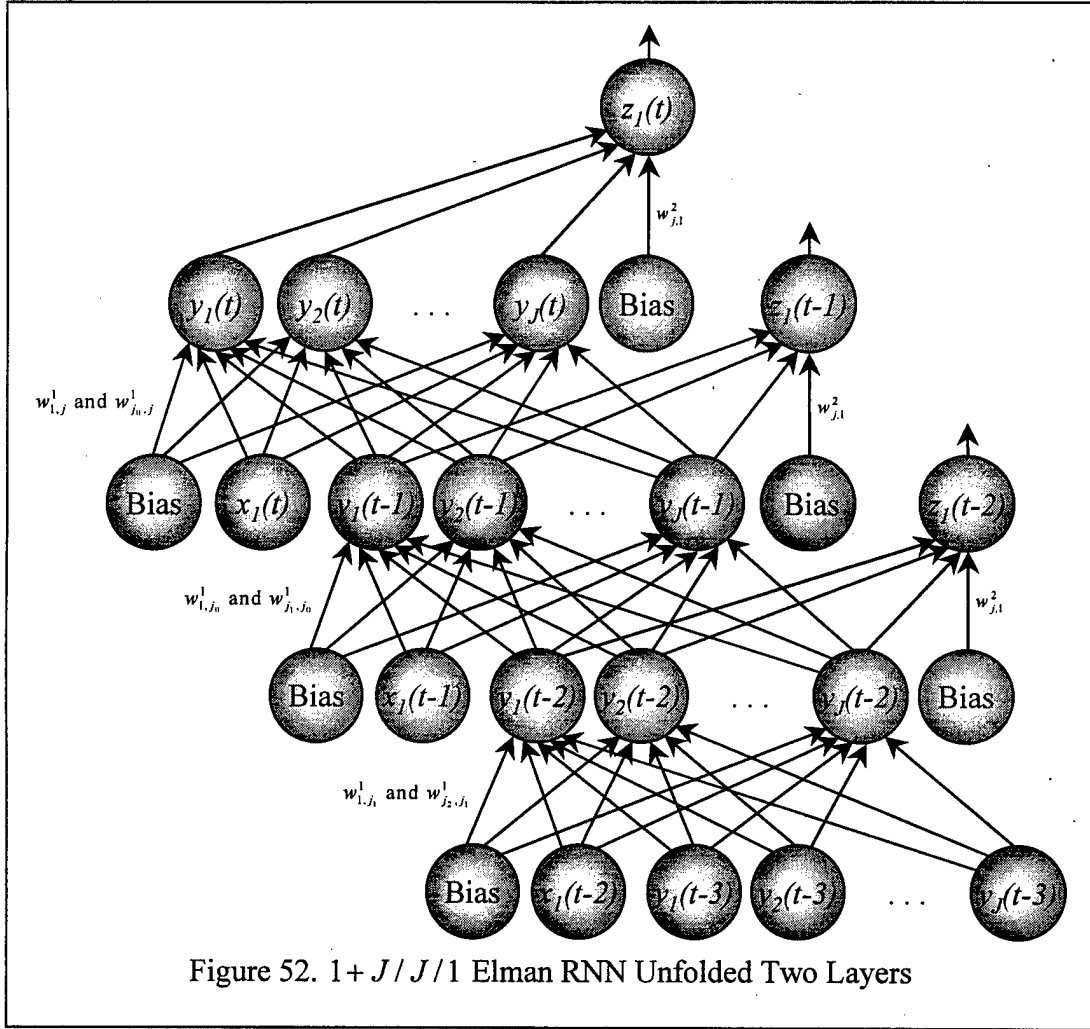
$$\Gamma_{y_{j_2}}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \dot{f}_1^2 \left(a_1^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1, \mathbf{W}) \right) \cdot \right.$$
$$\left. \sum_{j_1=1}^{J} w_{j_1,j_0}^1 \cdot \dot{y}_{j_1}(t-2, \mathbf{W}) \cdot w_{j_2,j_1}^1 \right|$$

(150)

8.2.8 N Time Lags for a $1 + J / J / 1$ Elman Recurrent Neural Network (RNN)

Continuing on with the unfolding, a $1 + J / J / 1$ Elman RNN can be unfolded N layers as shown in Figure 53. The partial derivative-based saliency measure can be calculated for the Nth unfolded layer by extending the equations in Sections 8.2.5

215

through 8.2.7. For the unfolded layer N in Figure 53, the partial derivative-based saliency measure for x_1 is calculated as:

$$\Gamma_{x_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial x_1(t-N)} \right| \tag{151}$$

More specifically:

$$\Gamma_{x_1}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \dot{f}_1^2 \left(a_1^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1, \mathbf{W}) \right) \cdot \right.$$
$$\left. \sum_{j_1=1}^{J} w_{j_1,j_0}^1 \cdot \dot{f}_{j_1}^1 \left(a_{j_1}^1(t-2, \mathbf{W}) \right) \cdot \ldots \cdot \sum_{j_{N-1}=1}^{J} w_{j_{N-1},j_{N-2}}^1 \cdot \dot{f}_{j_{N-1}}^1 \left(a_{j_{N-1}}^1(t-N, \mathbf{W}) \right) \cdot w_{1,j_{N-1}}^1 \right| \tag{152}$$

Similarly to Equation 151, the partial derivative-based saliency measure can be computed for context node y_{j_N} for $j_N = 1,2,\ldots J$ as:

$$\Gamma_{y_{j_N}}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \frac{\partial z_1(t, \mathbf{W})}{\partial y_{j_N}(t-N-1, \mathbf{W})} \right| \tag{153}$$

More specifically:

$$\Gamma_{y_{j_N}}^N = \frac{1}{T-N} \cdot \sum_{t=1}^{T-N} \left| \dot{f}_1^2 \left(a_1^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,1}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1, \mathbf{W}) \right) \cdot \right.$$
$$\left. \cdot \sum_{j_1=1}^{J} w_{j_1,j_0}^1 \cdot \dot{f}_{j_1}^1 \left(a_{j_1}^1(t-2, \mathbf{W}) \right) \cdot \ldots \cdot \sum_{j_{N-1}=1}^{J} w_{j_{N-1},j_{N-2}}^1 \cdot \dot{f}_{j_{N-1}}^1 \left(a_{j_{N-1}}^1(t-N, \mathbf{W}) \right) \cdot w_{j_N,j_{N-1}}^1 \right| \tag{154}$$

8.2.9   Derivations for a $I + J / J / K$ Elman Recurrent Neural Network (RNN)

The complexity of an Elman RNN architecture can be increased from a $1 + J / J / 1$ Elman RNN to a $I + J / J / K$ Elman RNN, the most general form of an Elman RNN, by allowing for I input nodes and K output nodes. Figure 54 shows a $I + J / J / K$ Elman RNN. The partial derivative-based saliency measure for a $1 + J / J / 1$ Elman RNN is calculated for feature x_i for $i = 1,2,\ldots,I$, using the training set
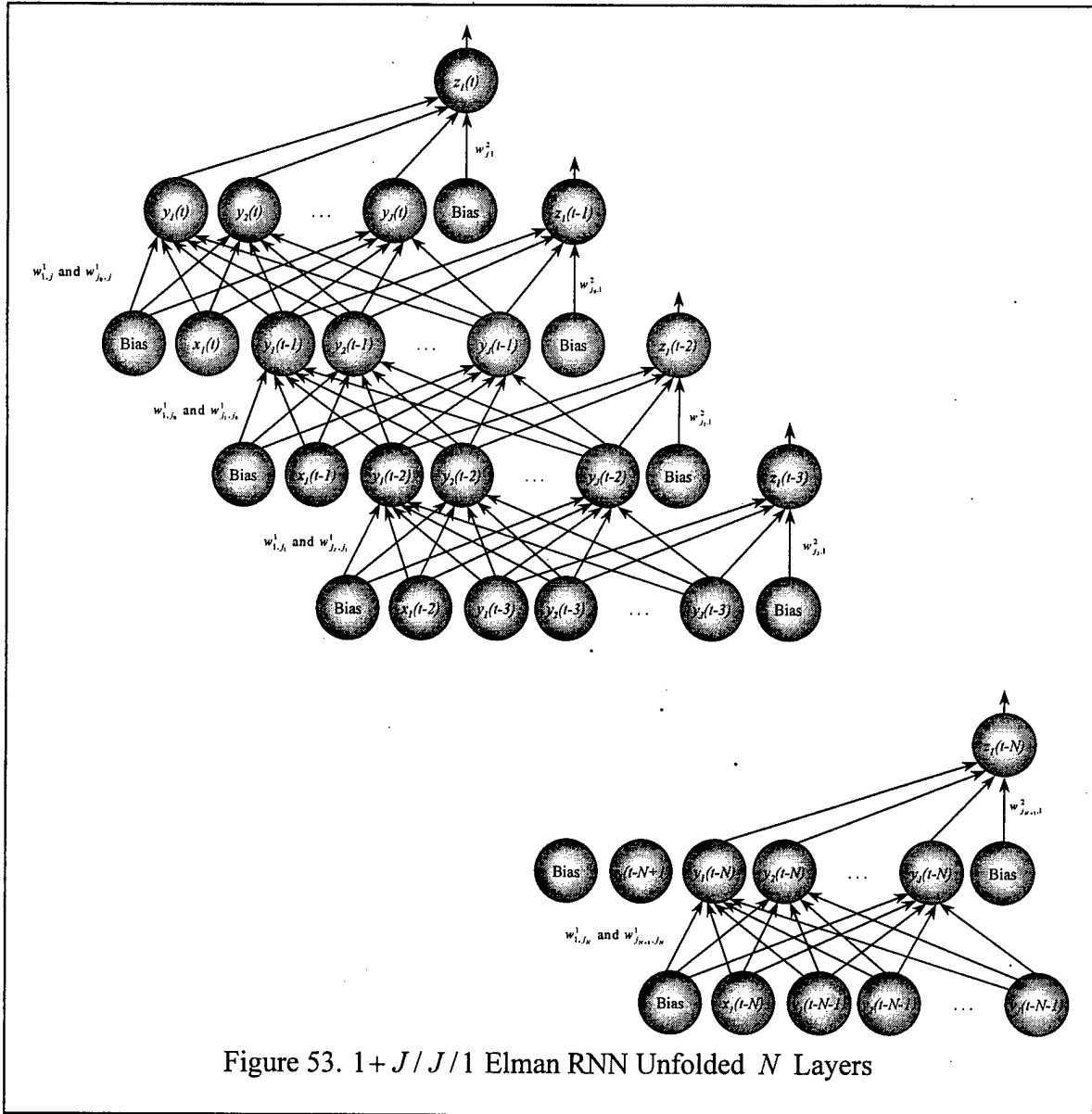
Figure 53. $1+J/J/1$ Elman RNN Unfolded N Layers

exemplars following Equation 119 as:

$$\Gamma_{x_i} = \frac{1}{K \cdot \mathrm{T}} \cdot \sum_{k=1}^{K} \sum_{t=1}^{\mathrm{T}} \left| \frac{\partial z_k(t, \mathbf{W})}{\partial x_i(t)} \right| \tag{155}$$

The main difference between Equation 155 and Equations 119 and 139 is the summation

over the output nodes. The summation difference will hold for every unfolded layer as

Figure 54. $I + J / J / K$ Elman RNN
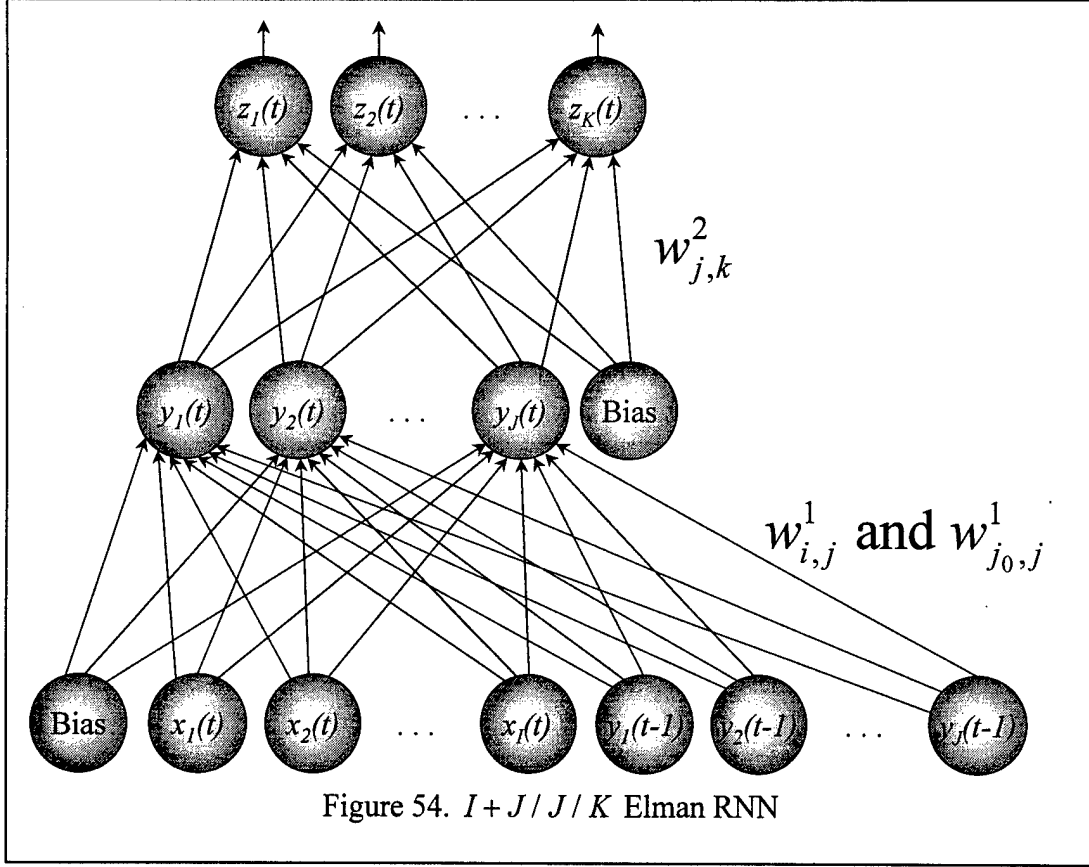
described in Sections 8.2.10 through 8.2.12. Equation 155 can be written more specifically as:

$$\Gamma_{x_i} = \frac{1}{K} \cdot \frac{1}{T} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T} \left| \dot{f}_k^2 \left(a_k^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot w_{i,j}^1 \right| \tag{156}$$

Similarly to Equation 156, the partial derivative-based saliency measure can be computed for context node y_{j_0} for $j_0 = 1, 2, \ldots J$ as:

$$\Gamma_{y_{j_0}} = \frac{1}{K \cdot T} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T} \left| \frac{\partial z_k(t, \mathbf{W})}{\partial y_{j_0}(t-1, \mathbf{W})} \right| \tag{157}$$

More specifically:

$$\Gamma_{y_{j_0}} = \frac{1}{K \cdot T} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T} \left| \dot{f}_k^2 \left(a_k^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot w_{j_0,j}^1 \right| \tag{158}$$

218

8.2.10 One Time Lag of a $I + J / J / K$ Elman Recurrent Neural Network (RNN)

A $I + J / J / K$ Elman RNN can be unfolded as shown in Figure 55. The partial derivative-based saliency measure can be calculated for the first unfolded layer by extending the equations in Section 8.2.9. For the unfolded layer 1 in Figure 55, the partial derivative-based saliency measure for feature x_i for $i = 1, 2, \ldots, I$ is computed as:

$$\Gamma_{x_i}^1 = \frac{1}{K \cdot (T-1)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left| \frac{\partial z_k(t, \mathbf{W})}{\partial x_i(t-1)} \right| \tag{159}$$

More specifically:

$$\Gamma_{x_i}^1 = \frac{1}{K \cdot (T-1)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left| \dot{f}_k^2 \left(a_k^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1, \mathbf{W}) \right) \cdot w_{i,j_0}^1 \right| \tag{160}$$

Similarly to Equation 159, the partial derivative-based saliency measure can be computed

Figure 55. $I + J / J / K$ Elman RNN Unfolded One Layer
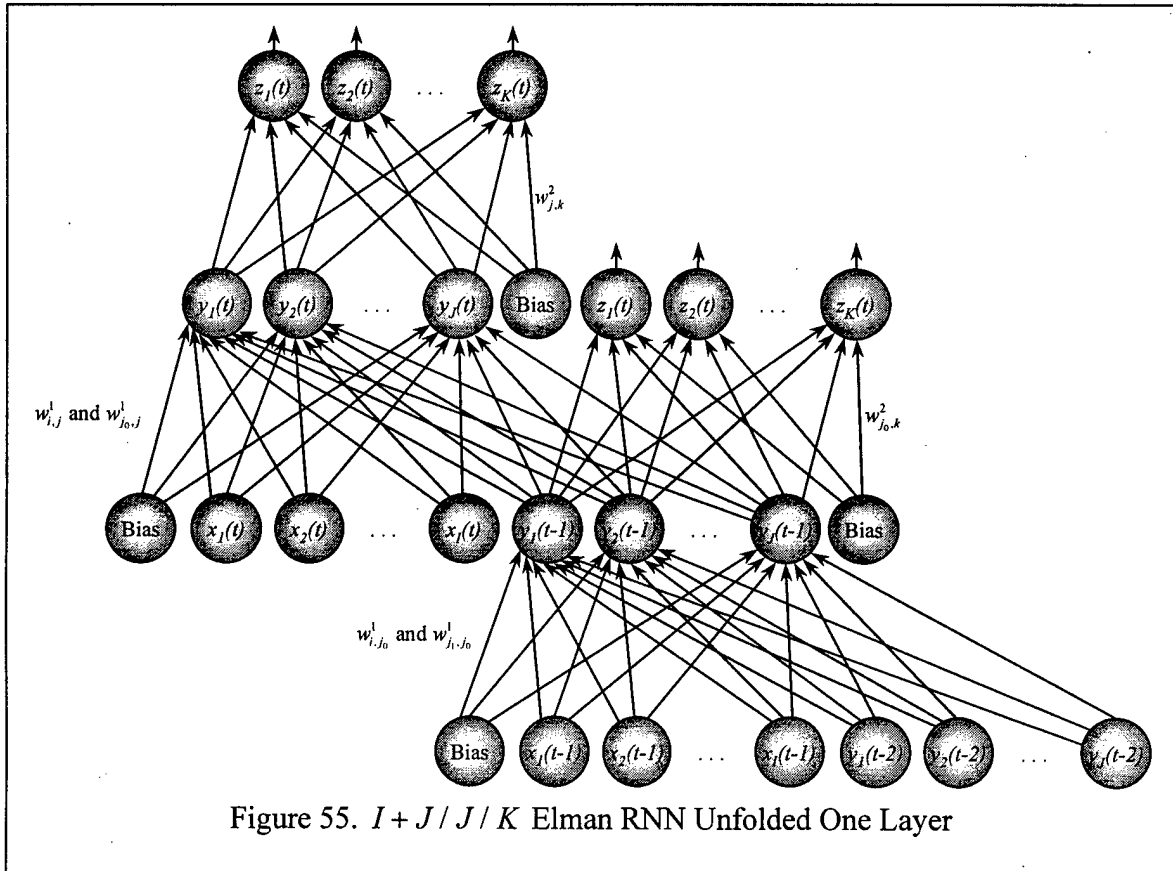
for context node y_{j_1} for $j_1 = 1,2,...J$ as:

$$\Gamma^1_{y_{j_1}} = \frac{1}{K \cdot (T-1)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-1} \left| \frac{\partial z_k(t,\mathbf{W})}{\partial y_{j_1}(t-2,\mathbf{W})} \right| \tag{161}$$

More specifically:

$$\Gamma^1_{y_{j_1}} = \frac{1}{K \cdot (T-1)} \sum_{k=1}^{K} \cdot \sum_{t=1}^{T-1} \left| \dot{f}^2_k \left(a^2_k(t,\mathbf{W}) \right) \cdot \sum_{j=1}^{J} w^2_{j,k} \cdot \dot{f}^1_j \left(a^1_j(t,\mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w^1_{j_0,j} \cdot \dot{f}^1_{j_0} \left(a^1_{j_0}(t-1,\mathbf{W}) \right) \cdot w^1_{j_1,j_0} \right| \tag{162}$$

8.2.11 Two Time Lags of a $I + J / J / K$ Elman Recurrent Neural Network (RNN)

The unfolding of an Elman RNN can be continued further. Figure 56 shows a $I + J / J / K$ Elman RNN unfolded two layers. The partial derivative-based saliency measure can be calculated for the second unfolded layer by extending the equations in Section 8.2.9 and Section 8.2.10. For the unfolded layer 2 in Figure 56, the partial derivative-based saliency measure for feature x_i for $i = 1,2,...,I$ is calculated as:

$$\Gamma^2_{x_i} = \frac{1}{K \cdot (T-2)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-2} \left| \frac{\partial z_k(t,\mathbf{W})}{\partial x_i(t-2)} \right| \tag{163}$$

More specifically:

$$\Gamma^2_{x_i} = \frac{1}{K \cdot (T-2)} \sum_{k=1}^{K} \sum_{t=1}^{T-2} \left| \dot{f}^2_k \left(a^2_k(t,\mathbf{W}) \right) \cdot \sum_{j=1}^{J} w^2_{j,k} \cdot \dot{f}^1_j \left(a^1_j(t,\mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w^1_{j_0,j} \cdot \dot{f}^1_{j_0} \left(a^1_{j_0}(t-1,\mathbf{W}) \right) \cdot \right.$$
$$\left. \sum_{j_1=1}^{J} w^1_{j_1,j_0} \cdot \dot{f}^1_{j_1} \left(a^1_{j_1}(t-2,\mathbf{W}) \right) \cdot w^1_{i,j_1} \right| \tag{164}$$

Similarly to Equation 163, the partial derivative-based saliency measure can be computed for context node y_{j_2} for $j_2 = 1,2,...J$ as:

$$\Gamma^2_{y_{j_2}} = \frac{1}{K \cdot (T-2)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-2} \left| \frac{\partial z_k(t,\mathbf{W})}{\partial y_{j_2}(t-3,\mathbf{W})} \right| \tag{165}$$

220

More specifically:

$$\Gamma^2_{y_{j_2}} = \frac{1}{K \cdot (T-2)} \sum_{k=1}^{K} \sum_{t=1}^{T-2} \left| \dot{f}^2_k \left(a^2_k (t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w^2_{j,k} \cdot \dot{f}^1_j \left(a^1_j (t, \mathbf{W}) \right) \cdot \sum_{j_0=1}^{J} w^1_{j_0,j} \cdot \dot{f}^1_{j_0} \left(a^1_{j_0} (t-1, \mathbf{W}) \right) \cdot \right.$$

$$\left. \cdot \sum_{j_1=1}^{J} w^1_{j_1,j_0} \cdot \dot{f}^1_{j_1} \left(a^1_{j_1} (t-2, \mathbf{W}) \right) \cdot w^1_{j_2,j_1} \right| \tag{166}$$

8.2.12 N Time Lags for a $I + J / J / K$ Elman Recurrent Neural Network (RNN)

Continuing on with the unfolding, a $I + J / J / K$ Elman RNN can be unfolded N layers as shown in Figure 57. The partial derivative-based saliency measure can be calculated for the Nth unfolded layer by extending the equations in Sections 8.2.9 through 8.2.11. For the unfolded layer N in Figure 57, the partial derivative-based saliency measure for feature x_i for $i = 1, 2, \ldots, I$ is calculated as:

$$\Gamma^N_{x_i} = \frac{1}{K \cdot (T-N)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-N} \left| \frac{\partial z_k (t, \mathbf{W})}{\partial x_i (t-N)} \right| \tag{167}$$

More specifically:

$$\Gamma^N_{x_i} = \frac{1}{K \cdot (T-N)} \sum_{k=1}^{K} \sum_{t=1}^{T-N} \left| \dot{f}^2_k \left(a^2_k (t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w^2_{j,k} \cdot \dot{f}^1_j \left(a^1_j (t, \mathbf{W}) \right) \cdot \right.$$

$$\sum_{j_0=1}^{J} w^1_{j_0,j} \cdot \dot{f}^1_{j_0} \left(a^1_{j_0} (t-1, \mathbf{W}) \right) \cdot \sum_{j_1=1}^{J} w^1_{j_1,j_0} \cdot \dot{f}^1_{j_1} \left(a^1_{j_1} (t-2, \mathbf{W}) \right) \cdot \ldots \cdot \tag{168}$$

$$\left. \sum_{j_{N-1}=1}^{J} w^1_{j_{N-1},j_{N-2}} \cdot \dot{f}^1_{j_{N-1}} \left(a^1_{j_{N-1}} (t-N, \mathbf{W}) \right) \cdot w^1_{i,j_{N-1}} \right|$$

Similarly to Equation 167, the partial derivative-based saliency measure can be computed for context node y_{j_N} for $j_N = 1, 2, \ldots J$ as:
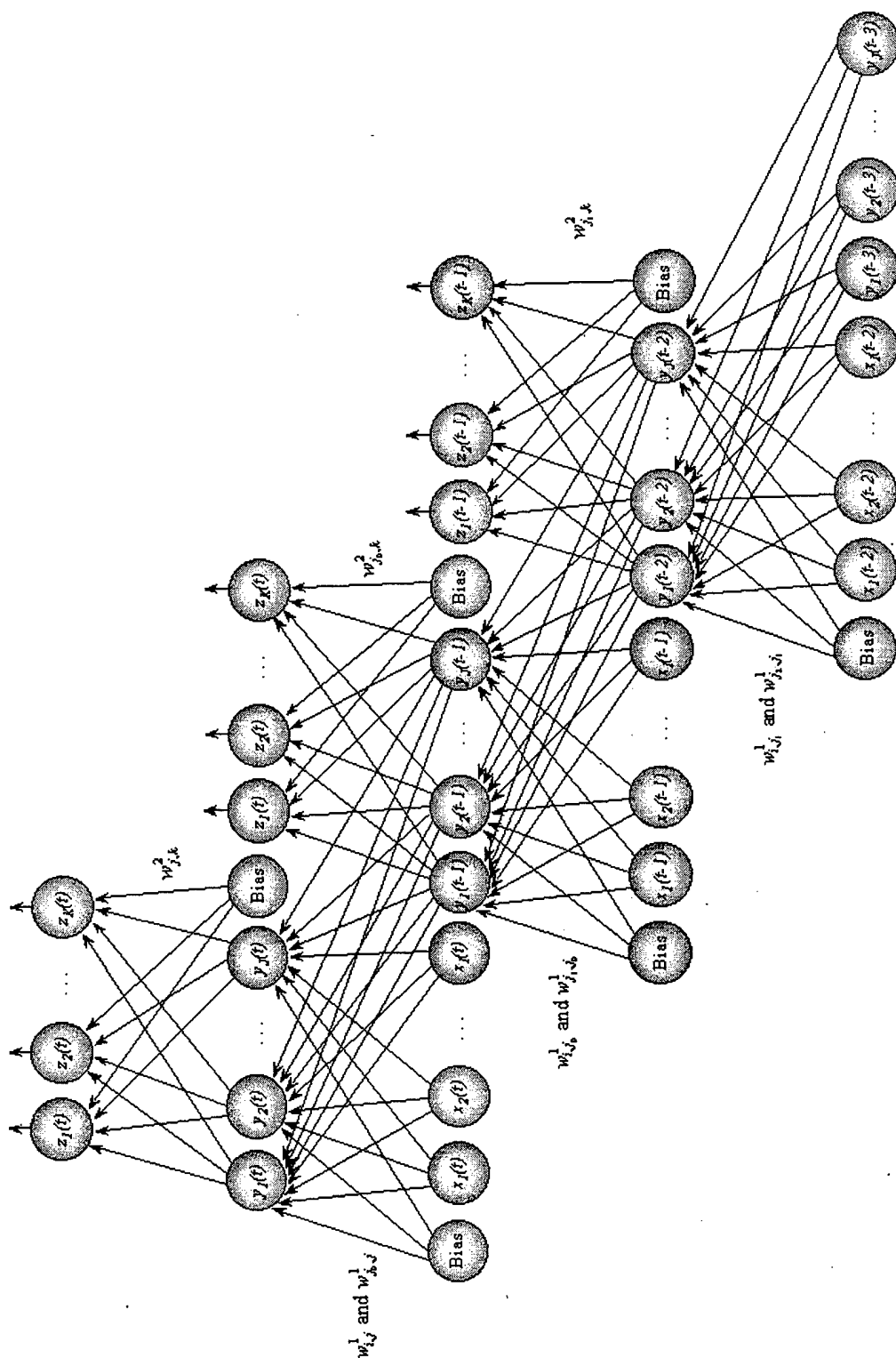
Figure 56. $I + J / J / K$ Elman RNN Unfolded Two Layers
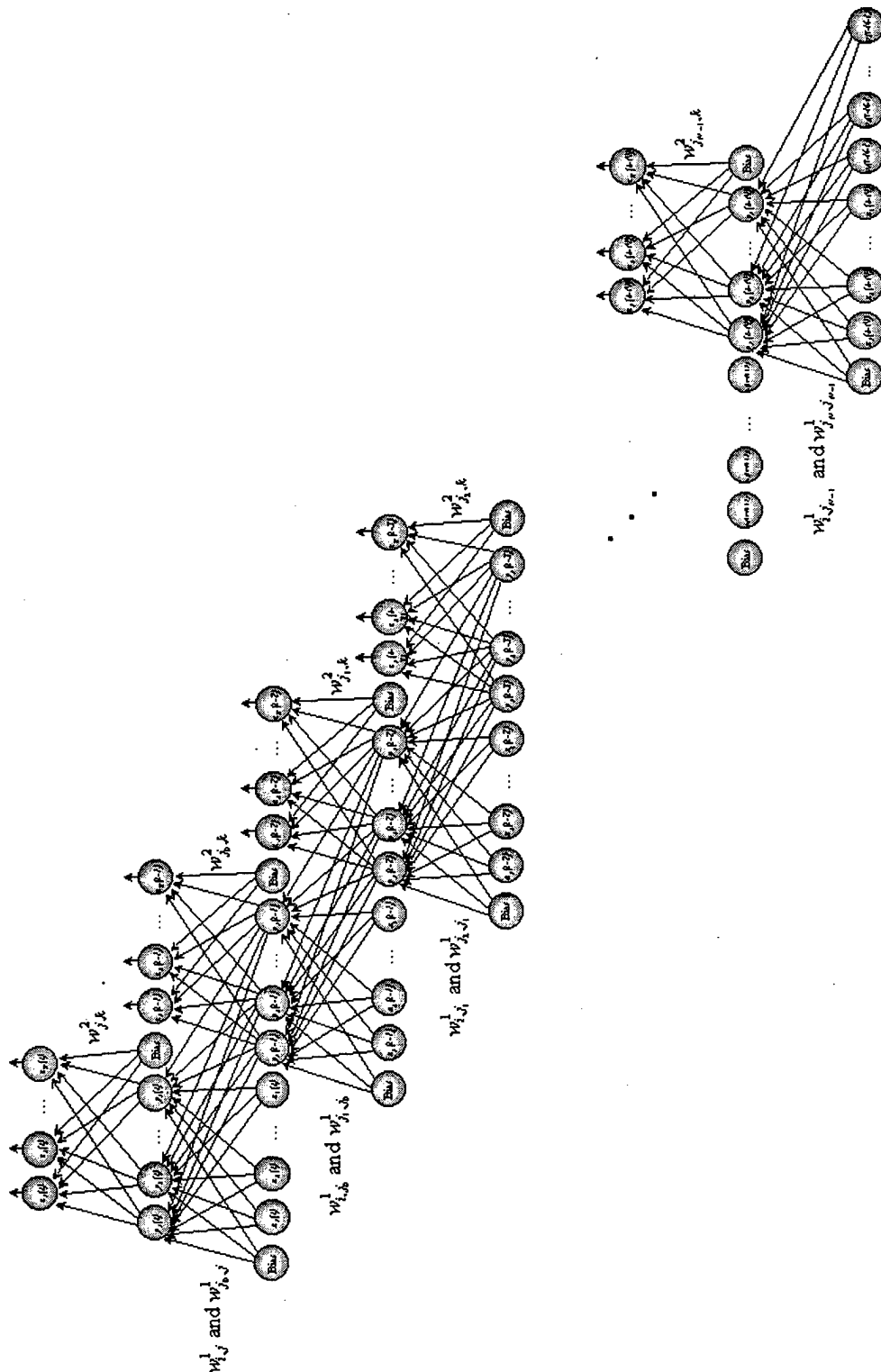
Figure 57. $I + J / J / K$ Elman RNN Unfolded N Layers

$$\Gamma^N_{y_{j_N}} = \frac{1}{K \cdot (T - N)} \cdot \sum_{k=1}^{K} \sum_{t=1}^{T-N} \left| \frac{\partial z_k(t, \mathbf{W})}{\partial y_{j_N}(t - N - 1, \mathbf{W})} \right| \tag{169}$$

More specifically:

$$\Gamma^N_{y_{j_N}} = \frac{1}{K \cdot (T - N)} \sum_{k=1}^{K} \sum_{t=1}^{T-N} \left| \dot{f}_k^2 \left(a_k^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{J} w_{j,k}^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot \right.$$

$$\sum_{j_0=1}^{J} w_{j_0,j}^1 \cdot \dot{f}_{j_0}^1 \left(a_{j_0}^1(t-1, \mathbf{W}) \right) \cdot \sum_{j_1=1}^{J} w_{j_1,j_0}^1 \cdot \dot{f}_{j_1}^1 \left(a_{j_1}^1(t-2, \mathbf{W}) \right) \cdot \ldots \cdot \tag{170}$$

$$\left. \sum_{j_{N-1}=1}^{J} w_{j_{N-1},j_{N-2}}^1 \cdot \dot{f}_{j_{N-1}}^1 \left(a_{j_{N-1}}^1(t-N, \mathbf{W}) \right) \dot{y}_{j_{N-1}}(t-N, \mathbf{W}) \cdot w_{j_N,j_{N-1}}^1 \right|$$

8.3 Spatial-Temporal Feature Screening Method

A spatial-temporal feature screening method to determine the parsimonious set of salient features in an Elman RNN was developed. The significant contribution of this spatial-temporal feature screening method is that it accounts for the temporal dimension of the features in addition to the spatial dimension. The spatial-temporal feature screening method utilizes the partial derivative-based spatial-temporal saliency measure derived in previous sections of the this Chapter to screen features.

The screening method is similar to the SNR screening method in that a noise feature denoted $x_N(t)$ for $t = 1, 2, \ldots, T$ is added to the set of candidate input features. The time series of noise is generated from a Uniform(0,1) random distribution. The spatial-temporal screening method is a backwards screening method that provides a mechanism to potentially identify a parsimonious set of salient features in both time and space. The screening method strives to maintain good generalization while removing non-salient features.

Spatial-Temporal Screening Method

1. Introduce a Uniform(0,1) noise feature $x_N(t)$ for $t = 1,2,\ldots,T$ to the original set of features.

2. Preprocess all features following Equation 16 or Equation 19.

3. Initialize the weights following the Nguyen-Widrow method [102].

4. Initialize all context nodes to 0.0.

5. Compute the fractal dimension of each input feature by the Grassberger and Procaccia method described in Section 2.5.1.1.

6. Apply Taken's Theorem in Equation 57 to determine the lag upper bound denoted as ℓ_{max}.

7. Set $\ell = 0$.

8. Train the Elman RNN for a pre-defined number of epochs. Keep the weights that minimize the MSE_{test}.

9. Compute CA_{test}.

10. Compute Γ_{x_i} for $i = 1,2,\ldots,I$ and Γ_{x_N}.

11. Set $\ell = \ell + 1$.

12. Compute $\Gamma_{x_i}^{\ell}$ for $i = 1,2,\ldots,I$.

13. If $\Gamma_{x_i}^{\ell} \leq \Gamma_{x_N}$ for $i = 1,2,\ldots,I$ or if $\ell = \ell_{max}$, set $\ell = \ell_{stop}$ and go to Step 14. Else go to Step 11.

14. Plot $\Gamma_{x_i}^{\ell}$ for $i = 1,2,\ldots,I$ versus $\ell = 0,1,2,\ldots,\ell_{stop}$.

15. Using the trapezoidal rule, compute the area under the *spatial-temporal saliency curve* for each feature x_i for $i = 1,2,\ldots,I$.

16. Remove the feature with the smallest area and set $I = I - 1$.

17. If $I > 0$, go to step 7. Else go to step 18.

18. Keep the set of salient features that produced the best CA_{test}. Remove $\dot{x}_N(t)$ for $t = 1, 2, \ldots, T$.

19. Initialize the weights following the Nguyen-Widrow method [102].

20. Initialize all context nodes to 0.0.

21. Train the Elman RNN with the set of salient features that produced the best CA_{test} until the MSE_{test} is minimized.

8.4 Application to Classifying Pilot Workload

8.4.1 Introduction

The spatial-temporal feature screening method was applied to a two-class and a three-class pilot workload problem. The objective of the two-class problem was to determine whether the pilot was in visual flight rules (VFR) or instrument flight rules (IFR) meteorological conditions. The objective of the three-class problem was to determine if the pilot's workload was low, medium, or high. Features for both classification problems were number of eye blinks, heart rate, and respiration rate. Data were collected from the test subject, a general aviation pilot with an instrument rating, flying a Piper Arrow single-engine airplane in both VFR and IFR conditions. Several types of ANNs (feedforward MLP ANN, TDNN, and Elman RNN) and several number of hidden nodes ($J = 1$, 2, 3, 4, 8, and 12 for the feedforward MLP ANN, $J = 1$, 2, 3, 4, 8, and 12 for the TDNN, and $J = 1$, 2, 3, and 4 for the Elman RNN) were utilized in an experimental design approach to feature screening. The SNR feature screening method was used in a feedforward MLP ANN and a TDNN. The new spatial-temporal feature screening method was used in an Elman RNN.

8.4.2   Data

The pilot workload data used in this chapter was actual flight data collected in both VFR and IFR conditions. An instrumented rated private pilot flew two sorties each on two separate days. Data collected during the first sortie was used for training. Data collected during the second sortie was used for testing. Since a third sortie was not conducted, no validation was conducted. The two sorties used in this dissertation each had 16 usable segments of flight described in Table 28. Other segments of flight were flown, but were not useable due mainly to data drop-out. For both sorties, the segments of flight were flown in the exact same order. Each segment lasted approximately two minutes. After completion of each two-minute segment of flight, the test subject pilot was asked to rate his workload for that segment between 0.0 and 100.0 with 100.0 being the highest. The average of the ratings for each of the two sorties was used to cluster the workload into low, medium, and high classifications.

For each segment of flight, several peripheral psychophysiological measurements were taken. EOG provided the number of eye blinks over 10-second moving windows with 50% overlap. EKG provided the average heart rate over 10-second moving windows with 50% overlap. Respiratory gauges provided the average respiration rate over 10-second moving windows with 50% overlap. Plots of the number of eye blinks, heart rate, and respiration rate are shown in Figure 58 for the training and test set. Each of the plots in Figure 58 are divided up into 16 blocks where each block represents a segment. The workload associated with each segment is denoted above each block where "L" is for low, "M" is for medium, and "H" is for high. The average over each segment is also provided in the plots in Figure 58 as a straight horizontal line. The three plots on

the left hand side of Figure 59 show the average over each segment for each feature for both the training and test sets.

In this dissertation, features were typically preprocessed by normalizing the features between 0.0 and 1.0 following Equation 19 by combining all data available in the training and test sets. But, the plots on the left hand side of Figure 59 for heart rate and respiration show that a day-to-day difference existed. In an attempt to account for the day-to-day differences, the first attempt at preprocessing the features was to normalize the features within each day. The three plots on the right hand side of Figure 59 show the average over each segment for each normalized feature for both the training and test sets. Unfortunately, this did not alleviate the day-to-day difference for respiration rate as shown in the respiration rate plot on the right hand side of Figure 59.

The next attempt at preprocessing the features was to standardize the features within each day following Equation 16. After standardization, the preprocessed features

Table 28. Segments of Flight

Segment	VFR/IFR	Description	Avg Rating	Workload
1	VFR	Take Off	50.0	Medium
2	VFR	Climb Out	42.5	Low
3	VFR	Cruise	45.0	Low
4	VFR	Air Work	42.5	Low
5	VFR	Approach	45.0	Low
6	VFR	Touch and Go	52.5	Medium
7	VFR	Climb Out	42.5	Low
8	IFR	Hold	57.5	High
9	IFR	DME Arc	62.5	High
10	IFR	ILS Tracking	72.5	High
11	IFR	Missed Approach	55.0	Medium
12	IFR	Climb Out	47.5	Medium
13	IFR	High Speed Hold	57.5	High
14	IFR	High Speed DME Arc	55.0	Medium
15	IFR	High Speed ILS Tracking	62.5	High
16	IFR	Landing	60.0	High

have 0.0 mean and unit variance. Plots of the number of eye blinks, heart rate, and respiration rate after standardization are shown in Figure 60 for the training and test set. Like Figure 58, the plots in Figure 60 are divided up into 16 blocks where each block represents a segment. The workload associated with each segment is denoted above each block where "L" is for low, "M" is for medium, and "H" is for high. The average over each segment is also provided in the plots in Figure 60 as a straight horizontal line. The three plots on the right hand side of Figure 61 show the average over each segment for each standardized feature for both the training and test sets. As a reference, the plots on the left hand side of Figure 61 are the same as the plots on the left hand side of Figure 59. It appears from the plots on the right hand side of Figure 61 that preprocessing the input features via standardization within each day accounted for the day-to-day differences. Throughout the experimental design, all features were standardized within each day.

8.4.3 Methodology

The objective was to apply a feature screening method to three peripheral psychophysiological features used to classify pilot workload in order to determine the parsimonious set of salient features for three types of ANNs:

1. Feedforward MLP ANN
2. TDNN
3. Elman RNN.

For each type of ANN used, several different numbers of hidden nodes were used. The SNR screening method was used for workload classification using feedforward MLP ANNs and TDNNs. The new spatial-temporal feature screening method was used for workload classification using Elman RNNs.
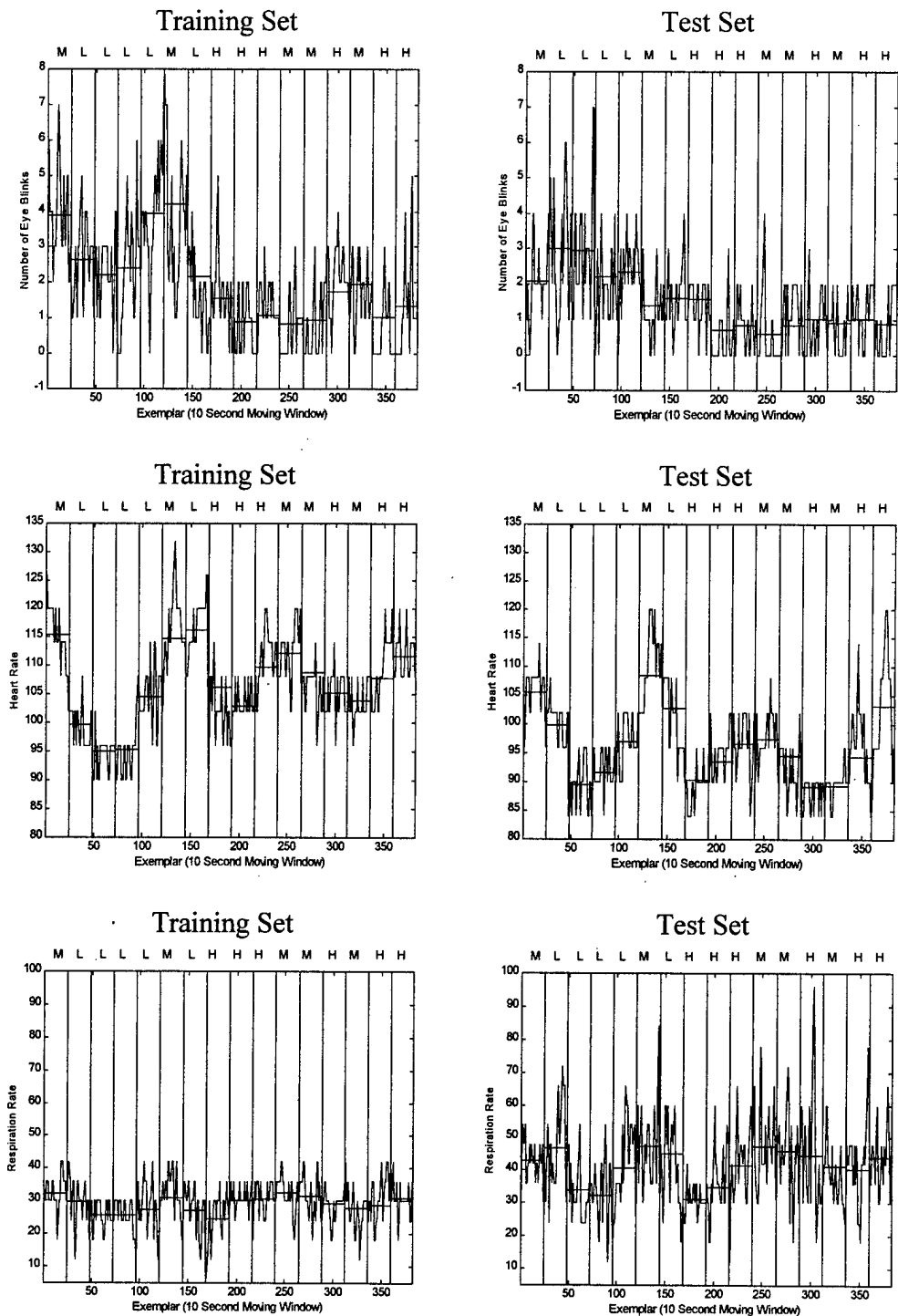
Figure 58. Data Averaged over 10-Second Moving Window with 50% Overlap

Figure 59. Data Averaged over Each Segment

231
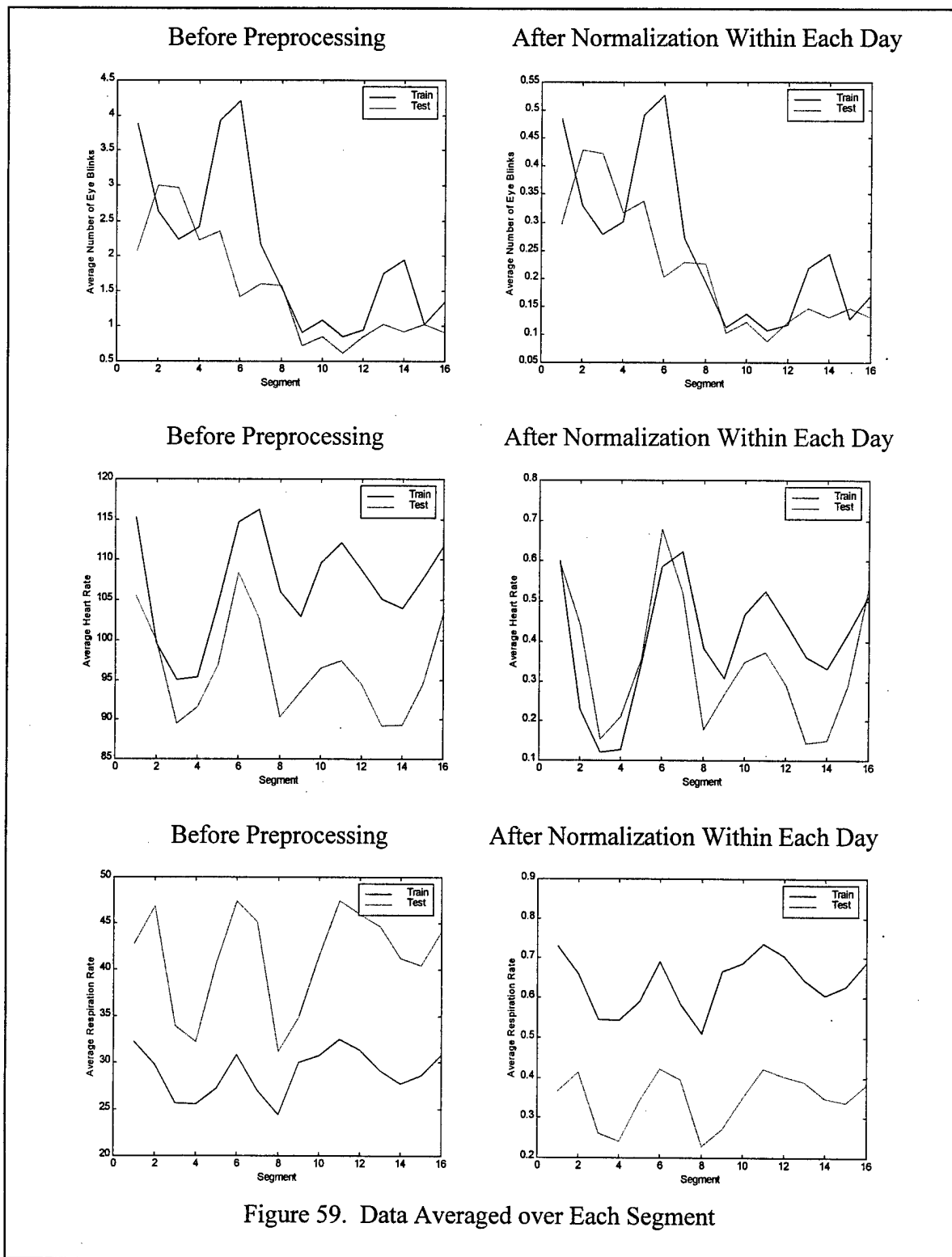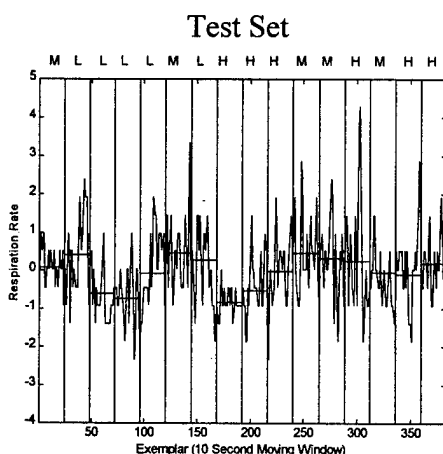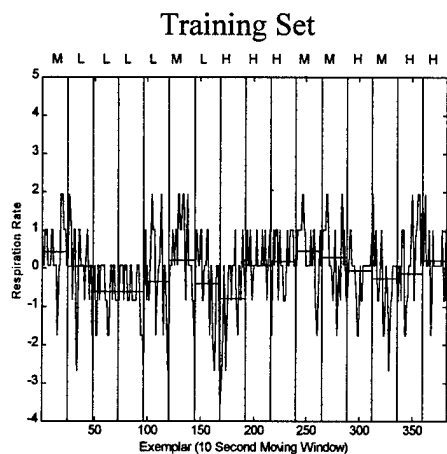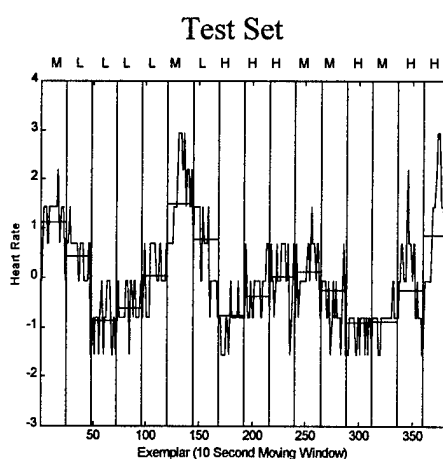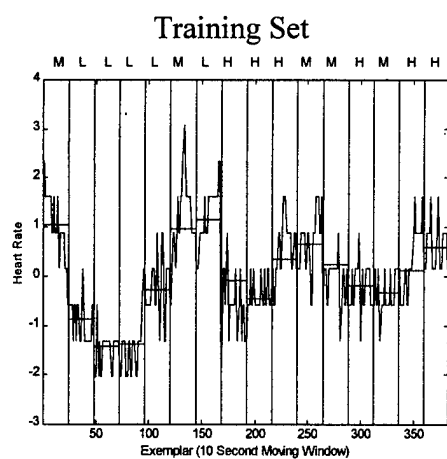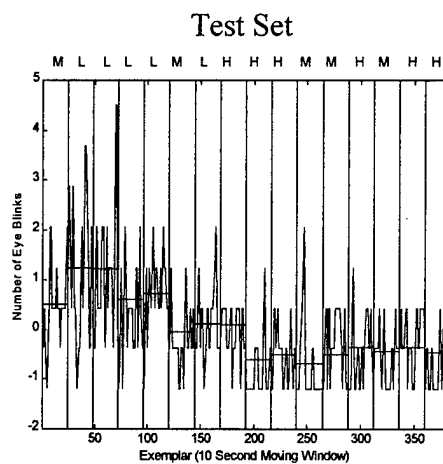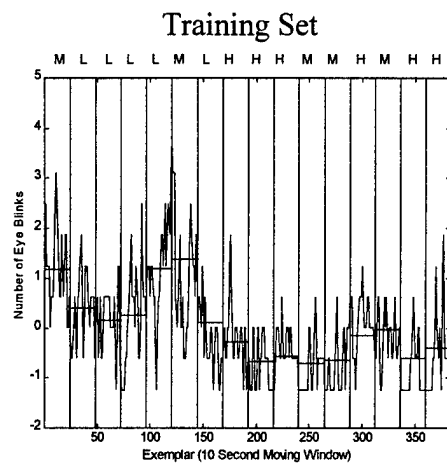
Figure 60. Standardized Data Averaged over 10-Second Moving Window with 50% Overlap
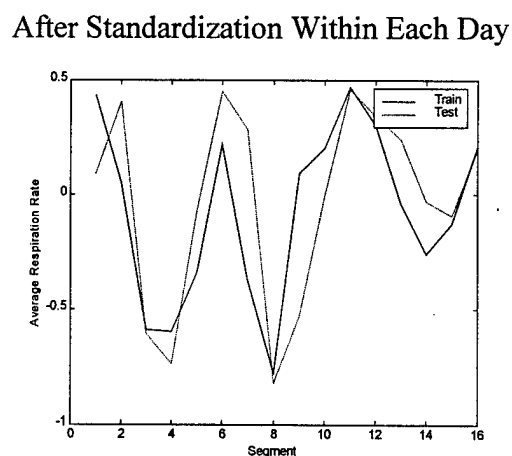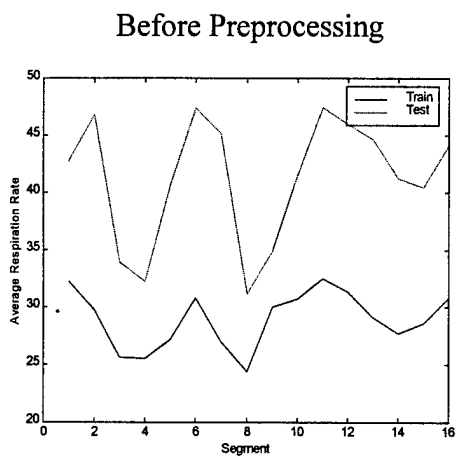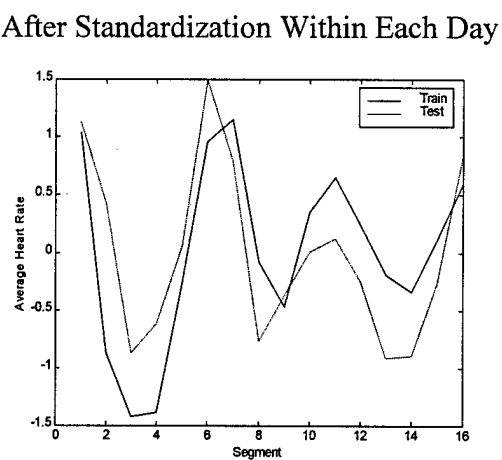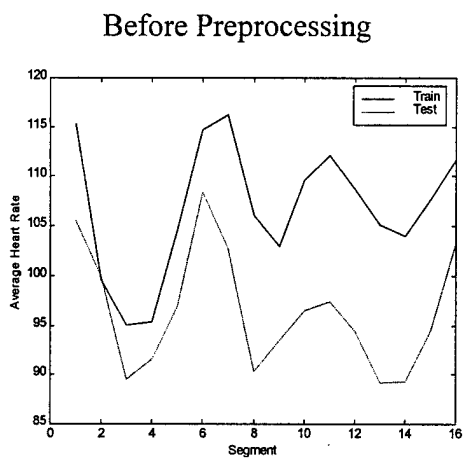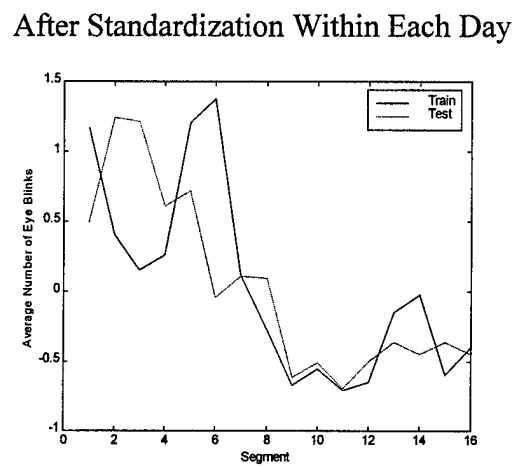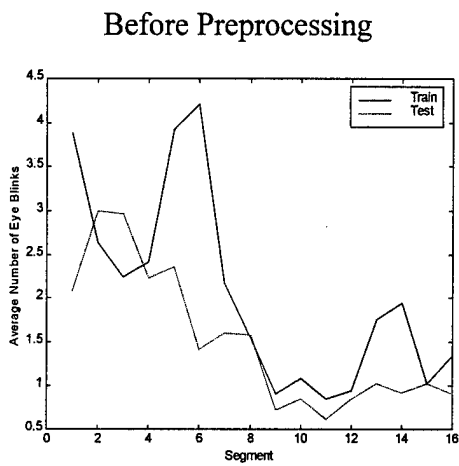
Figure 61. Standardized (0,1) Data Averaged over Each Segment

Two pilot workload classification problems were used. In the first, the objective of the ANN was to classify the pilot's workload as VFR or IFR (second column in Table 28). In the second, the objective of the ANN was to classify the pilot's workload as low, medium, or high (fifth column in Table 28). In order to compare performance, an experimental design was performed. There were three factors in the experimental design:

1. ANN type
   - Feedforward MLP ANN
   - TDNN
   - Elman RNN

2. Number of workload classes
   - VFR/IFR
   - Low/Medium/High

3. Number of hidden nodes
   - $J = 1$
   - $J = 2$
   - $J = 3$
   - $J = 4$
   - $J = 8$ (only for feedforward MLP ANN and TDNN)
   - $J = 12$ (only for feedforward MLP ANN and TDNN)

For all three types of ANNs, $K = 2$ output nodes were used for the two class pilot workload classification problem (VFR/IFR) and $K = 3$ output nodes were used for the three class pilot workload classification problem (low/medium/high). In the two class pilot workload classification problem, one output node corresponded to VFR and the other output node corresponded to IFR. The desired output for VFR exemplars was $\mathbf{z} = \begin{bmatrix} 1.0 & -1.0 \end{bmatrix}$. The desired output for IFR exemplars was $\mathbf{z} = \begin{bmatrix} -1.0 & 1.0 \end{bmatrix}$. For the three class pilot workload classification problem, one output node corresponded to low workload, one output node corresponded to medium workload, and one output corresponded to high workload. The desired output for low workload exemplars was

234

$\mathbf{z} = \begin{bmatrix} 1.0 & -1.0 & -1.0 \end{bmatrix}$. The desired output for medium workload exemplars was

$\mathbf{z} = \begin{bmatrix} -1.0 & 1.0 & -1.0 \end{bmatrix}$. The desired output for high workload exemplars was

$\mathbf{z} = \begin{bmatrix} -1.0 & -1.0 & 1.0 \end{bmatrix}$. The actual output of the ANN was determined using the *winner take all* strategy.

In all replications, the ANN was trained for 500 epochs using batch backpropagation with momentum and an adaptive learning rate following Equation 50. Momentum was implemented following the description given in Section 2.4.10. The adaptive learning rate was implemented following the description given in Section 2.4.11. The weights from the epoch that produced the minimum MSE_{test} were kept. All hidden, context, and output nodes were activated by the nonlinear hyperbolic tangent transfer function in Equation 6.

8.4.3.1 Feedforward Multilayer (MLP) Artificial Neural Network (ANN) Experimental Design

The SNR screening method was applied to a feedforward MLP ANN for $J = 1$, 2, 3, 4, 8, and 12 hidden nodes for the two-class and three-class pilot workload problems. For each level of J, the SNR screening method was replicated 30 times. For each replication of the SNR screening method, a random seed set equal to the replication number was used to initialize the weights. In other words, the random seed was set to 1 for the first replication and the random seed was set to 2 for the second replication and so on.

8.4.3.2 Time Delay Neural Network (TDNN) Experimental Design

In order to determine the maximum number of lags to use in a TDNN, the fractal

dimension was computed following the Grassberger and Procaccia algorithm described in

Section 2.5.1.1 for the following:

- Standardized injected noise feature in the training set
- Standardized injected noise feature in the test set
- Standardized number of eye blinks in the training set
- Standardized number of eye blinks in the test set
- Standardized heart rate in the training set
- Standardized heart rate in the test set
- Standardized respiration rate in the training set
- Standardized respiration rate in the test set.

In using the Grassberger and Procaccia algorithm, $k = 3, 4, 5, 6,$ and 7 and

$\ell = 0.6, 0.8, 1.0, 1.2, 1.4, 1.6,$ and 1.8. Taken's Theorem as given in Equation 57 was

then applied to determine the maximum number of lags required.

The SNR screening method was applied to a TDNN for $J = 1, 2, 3, 4, 8,$ and 12

hidden nodes for the two-class and three-class pilot workload classification problems.

For each level of J, the SNR screening method was replicated 30 times. For each

replication of the SNR screening method, a random seed set equal to the replication

number was used to initialize the weights. This was the first time that the SNR screening

method was applied to the lagged inputs of a TDNN. The SNR screening method was

allowed to remove any lag of any input feature. As an example, the SNR screening

method may remove $x_1(t-3)$ and $x_1(t-5)$ while retaining $x_1(t)$, $x_1(t-1)$, $x_1(t-2)$,

$x_1(t-4)$, and $x_1(t-6)$ for feature x_1 with $\ell_{max} = 6$. Further, the SNR screening method

may remove $x_2(t)$, $x_2(t-3)$, and $x_2(t-4)$ while retaining, $x_2(t-1)$, $x_2(t-2)$,

$x_2(t-5)$, and $x_2(t-6)$ for feature x_2 with $\ell_{max} = 6$. This is a novel approach to

236

utilizing a TDNN since in the past, all lagged inputs for all features were used as inputs to a TDNN. The CA_{test} may increase as a result of removing nonsalient lagged inputs.

8.4.3.3 Elman Recurrent Neural Network (RNN) Experimental Design

Finally, the spatial-temporal feature screening method was applied to an Elman RNN for $J = 1, 2, 3$, and 4 hidden nodes for the two-class and three-class pilot workload classification problems. The spatial-temporal feature screening method was not applied to an Elman RNN for $J = 8$ and 12 because results from the feedforward MLP ANN and TDNN replications showed that $J = 1, 2, 3$, and 4 hidden nodes were sufficient for both workload classification problems. For each level of J, the spatial-temporal feature screening method was replicated 30 times. For each replication of the spatial-temporal feature screening method, a random seed set equal to the replication number was used to initialize the weights.

8.4.4   Results

8.4.4.1   Feedforward Multilayer Perceptron (MLP) Artificial Neural Network (ANN) Experimental Design

Table 29 summarizes the average classification accuracy denoted as $\overline{CA}$ from applying the SNR screening method thirty times to a feedforward MLP ANN. The results are given for the $K = 2$ class workload problem (VFR/IFR) and the $K = 3$ class workload problem (low/medium/high). The various $\overline{CA}$s summarized in Table 29 include the average CA_{train} using all three features (and the injected noise) denoted as $\overline{CA}_{train}(3)$, the average CA_{test} using all three features (and the injected noise) denoted as

237

Table 29. $\overline{CA}$ Results from SNR Screening Method for Feedforward MLP ANN

K	J	$\overline{CA}_{train}(3)$	$\overline{CA}_{test}(3)$	$\overline{CA}_{test}(2)$	$\overline{CA}_{test}(1)$	$\overline{CA}_{test}(N)$
2	1	76.07%	66.48%	67.43%	70.17%	55.23%
	2	82.24%	66.97%	66.71%	58.13%	55.78%
	3	85.48%	66.65%	63.45%	51.75%	55.07%
	4	86.86%	66.20%	64.04%	52.38%	54.61%
	8	89.14%	66.48%	59.61%	53.38%	53.99%
	12	90.84%	66.76%	57.00%	54.31%	53.72%
3	1	48.80%	38.89%	39.67%	36.98%	36.36%
	2	59.07%	42.36%	42.24%	37.39%	36.35%
	3	62.46%	43.66%	43.05%	37.51%	36.79%
	4	64.38%	42.97%	43.31%	37.03%	36.83%
	8	68.51%	43.10%	40.50%	37.24%	36.05%
	12	71.69%	42.82%	40.70%	37.51%	35.71%

$\overline{CA}_{test}(3)$, the average CA_{test} using the top two salient features (and the injected noise)

denoted as $\overline{CA}_{test}(2)$, the average CA_{test} using the top salient feature (and the injected

noise) denoted as $\overline{CA}_{test}(1)$, and the average CA_{test} with noise as the only input denoted as

$\overline{CA}_{test}(N)$.

For each TDNN architecture, Table 30 lists the maximum CA_{test} denoted as

$\max(CA_{test})$ attained from applying the SNR screening method thirty times. In addition,

Table 30 lists the number of salient features I that produced the maximum CA_{test} in

addition to the associated CA_{train}.

For each feedforward MLP ANN architecture, Table 31 lists the parsimonious set

of salient features and their rankings resulting from the SNR screening method that

produced the maximum CA_{test} listed in Table 30. In Table 31, EB is for number of eye

blinks, HR is for heart rate, and RR is for respiration rate. Table 32 summarizes the

results attained after the feedforward MLP ANN was retrained 30 times using only the

Table 30. $\mathbf{max}(CA_{test})$ Results from SNR Screening Method
for Feedforward MLP ANN

K	J	I	$\mathbf{max}(CA_{test})$	CA_{train}
2	1	1	71.02%	76.24%
	2	2	72.06%	84.07%
	3	3	72.85%	85.64%
	4	1	71.02%	76.76%
	8	3	70.76%	89.82%
	12	3	71.28%	90.86%
3	1	3	50.91%	42.30%
	2	1	49.35%	45.95%
	3	3	46.74%	60.84%
	4	2	45.43%	64.49%
	8	2	48.04%	66.06%
	12	2	45.95%	66.06%

salient features without the injected noise feature. For both workload classification problems, there were 30 feedforward MLP ANNs trained using the architecture with the set of salient features that produced the maximum $\overline{CA_{test}}$ and 30 feedforward MLP ANNs trained using the architecture with the set of salient features that produced the maximum CA_{test}.

8.4.4.1.1 Visual Flight Rules (VFR) / Instrument Flight Rules (IFR) Classification Problem

For the VFR/IFR classification problem, the 1/1/2 feedforward MLP ANN architecture produced the maximum $\overline{CA_{test}} = 70.17\%$ as highlighted in Table 29. In all thirty replications of the SNR screening method as applied to a 1/1/2 feedforward MLP ANN, the only salient feature was the number of eye blinks.

Of the thirty replications of the SNR screening method as applied to $1/1/2$ feedforward MLP ANNs, the best replication produced $CA_{test} = 71.02\%$ as shown in Table 30. But, the replication that produced the overall highest $CA_{test} = 72.85\%$ was a $3/3/2$ feedforward MLP ANN that included all three peripheral psychophysiological features as highlighted in Table 30. Table 31 lists the set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the $1/1/2$ feedforward MLP ANN architecture, the only salient feature was number of eye blinks as highlighted in Table 31. For the $3/3/2$ feedforward MLP ANN architecture, the salient features in order were:

1. Heart rate
2. Respiration rate
3. Number of eye blinks

as highlighted in Table 31.

Thirty $1/1/2$ feedforward MLP ANNs with number of eye blinks as the only

Table 31. Results from SNR Screening Method for Feedforward MLP ANN

K	$J = 1$	$J = 2$	$J = 3$	$J = 4$	$J = 8$	$J = 12$
2	1. EB	1. HR	1. HR	1. EB	1. HR	1. HR
		2. EB	2. RR		2. RR	2. RR
			3. EB		3. EB	3. EB
3	1. EB	1. EB	1. HR	1. HR	1. HR	1. HR
	2. HR		2. EB	2. EB	2. EB	2. EB
	3. RR		3. RR			

Table 32. Best Results without Noise for Feedforward MLP ANN

K	J	I	$\overline{CA}_{test}$	$\overline{CA}_{train}$	$\max(CA_{test})$	CA_{train}
2	1	1	70.23%	76.24%	70.23%	76.24%
	3	3	64.83%	85.24%	68.14%	84.07%
3	1	3	38.43%	46.70%	43.86%	46.74%
	3	3	43.38%	63.68%	44.91%	65.80%

feature and thirty $3/3/2$ feedforward MLP ANNs with all three input features were trained. The results are shown in Table 32. For the VFR/IFR classification problem, the maximum $\overline{CA_{test}} = 70.23\%$ and the maximum $CA_{test} = 70.23\%$ was attained using only number of eye blinks in a $1/1/2$ feedforward MLP ANN. Figure 62 provides plots of the actual and desired outputs for the feedforward MLP ANNs listed in Table 32 for the VFR/IFR classification problem.

8.4.4.1.2 Low/Medium/High Workload Classification Problem

For the low/medium/high workload classification problem, the $3/3/3$ feedforward MLP ANN architecture produced the maximum $\overline{CA_{test}} = 43.66\%$ as highlighted in Table 29.

Of the thirty replications of the SNR screening method as applied to $3/3/3$ feedforward MLP ANNs, the best replication produced $CA_{test} = 46.74\%$ as shown in Table 30. But, the replication that produced the overall highest $CA_{test} = 50.91\%$ was a $3/1/3$ feedforward MLP ANN that included all three peripheral psychophysiological features as highlighted in Table 30. Table 31 lists the set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the $3/1/3$ feedforward MLP ANN architecture, the salient features in order were:

1. Number of eye blinks
2. Heart rate
3. Respiration rate

as highlighted in Table 31. For the $3/3/3$ feedforward MLP ANN architecture, the salient features in order were:

241

1. Heart rate
2. Number of eye blinks
3. Respiration rate

as highlighted in Table 31.

Thirty 3/1/3 feedforward MLP ANNs and thirty 3/3/3 feedforward MLP ANNs were trained without noise. The results are shown in Table 32. For the three-class pilot workload problem, the maximum $\overline{CA_{test}} = 43.38\%$ and the maximum $CA_{test} = 44.91\%$ was attained using all three peripheral psychophysiological features in a 3/3/3 feedforward MLP ANN. Figure 63 provides plots of the actual and desired

$$CA_{train} = 76.24\%$$

$$CA_{test} = 70.23\%$$

$$CA_{train} = 84.07\%$$

$$CA_{test} = 68.14\%$$
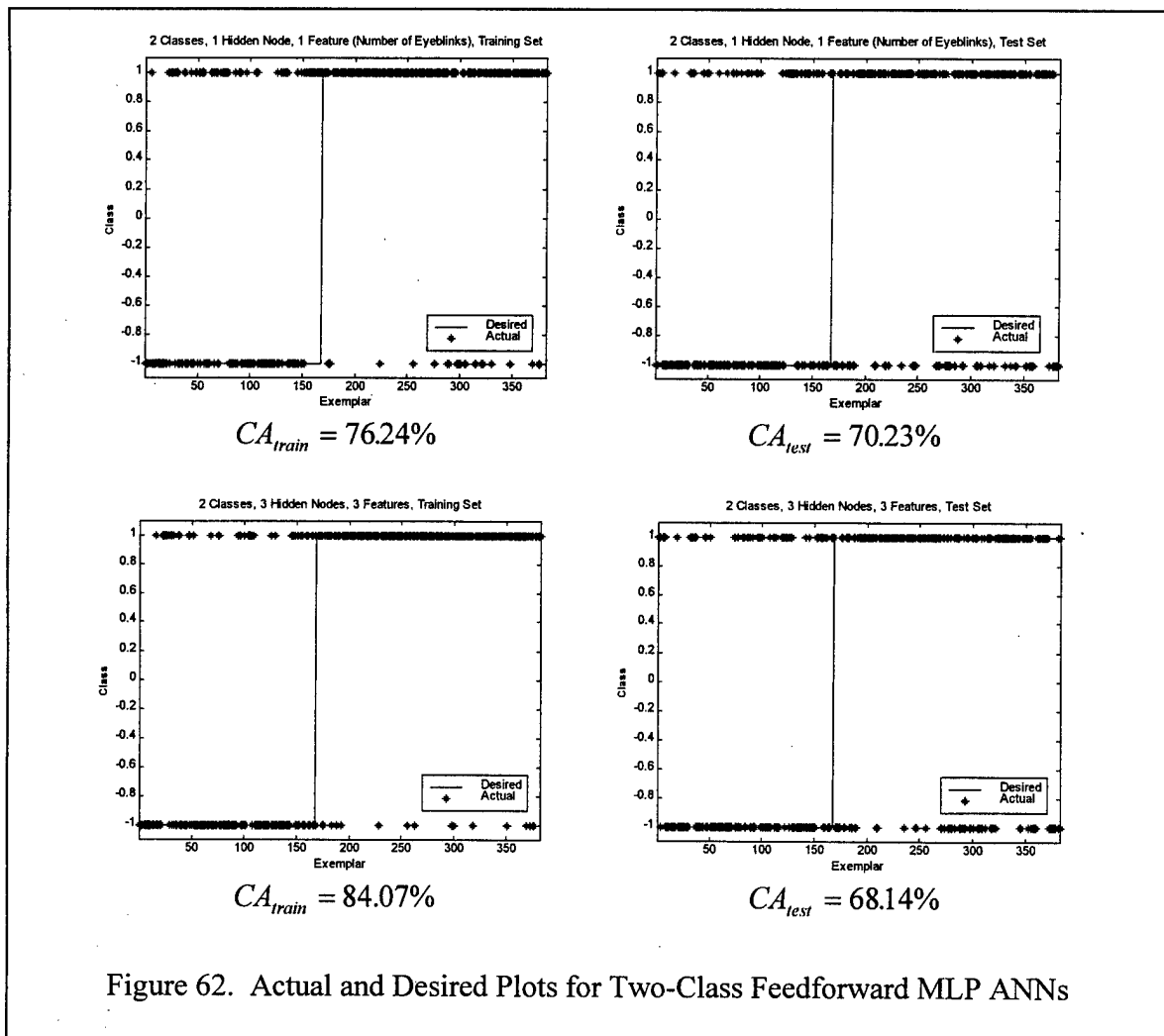
Figure 62. Actual and Desired Plots for Two-Class Feedforward MLP ANNs

242

outputs for the feedforward MLP ANNs listed in Table 32 for $K = 3$.

8.4.4.2 Time Delay Neural Network (TDNN) Experimental Design

The computed estimates of the fractal dimensions for the input features and the injected noise feature are shown in Table 33. The fractal dimension for the injected noise feature in the training set had the largest $f_d(A) = 4.5397$. Applying Taken's Theorem as in Equation 57 provided an upper bound to the maximum number of lags L so that:

$$CA_{train} = 46.74\%$$

$$CA_{test} = 43.86\%$$

$$CA_{train} = 65.80\%$$

$$CA_{test} = 44.91\%$$

Figure 63. Actual and Desired Plots for Three-Class Feedforward MLP ANNs
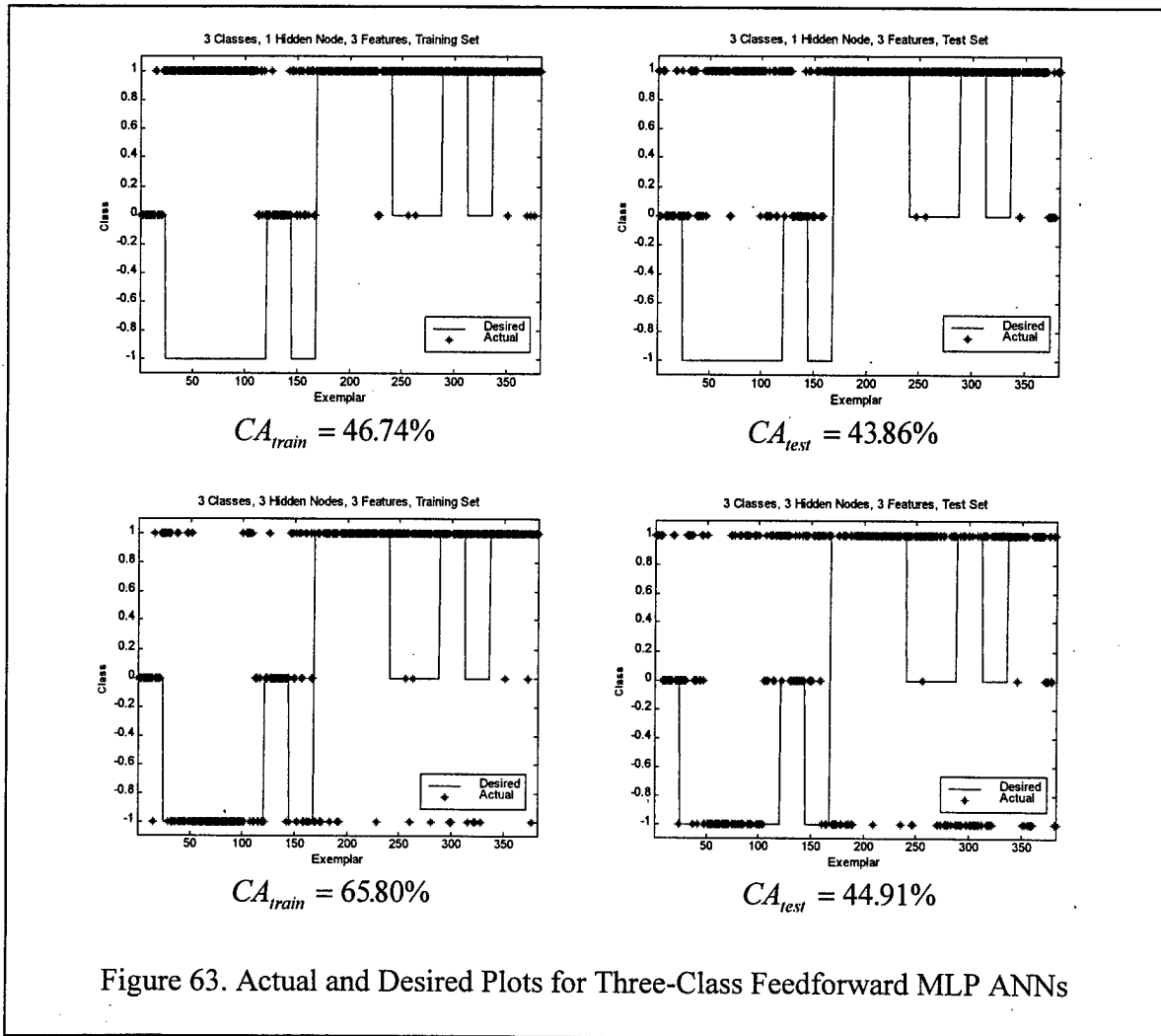
Table 33. Estimated Fractal Dimension of Injected Noise and Input Features

Feature	Training Set	Test Set
Noise	4.5397	4.4784
Number of Eye Blinks	3.4803	3.6120
Heart Rate	3.2991	3.3513
Respiration Rate	3.5586	4.1199

$$L + 1 < 2 \cdot f_d(A) + 1$$

$$L < 2 \cdot 4.5397$$

$$L < 9.0794$$

The maximum number of lag L used was 9. For each feature x_i for $i = 1,2,...,I$, the following nine lags were included: $x_i(t)$, $x_i(t-1)$, $x_i(t-2)$, $x_i(t-3)$, $x_i(t-4)$, $x_i(t-5)$, $x_i(t-6)$, $x_i(t-7)$, and $x_i(t-8)$. Note that the input $x_i(t)$ counts as one of the lags where $l = 0$. Since there were three peripheral psychophysiological features, there were a total of $3 \cdot 9 = 27$ input features to the TDNN in addition to one injected noise feature.

Table 34. $\overline{CA}_{test}$ Results from SNR Screening Method for TDNN

K	J	I	$\overline{CA}_{test}$	$\overline{CA}_{train}$
2	1	6	78.21%	85.30%
	2	8	79.53%	95.91%
	3	10	79.51%	99.46%
	4	11	80.12%	99.85%
	8	11	79.86%	99.99%
	12	16	78.80%	100.00%
3	1	16	42.87%	52.80%
	2	14	51.13%	73.94%
	3	15	50.39%	80.73%
	4	24	49.43%	87.50%
	8	26	47.41%	97.24%
	12	9	46.57%	92.22%

Table 35. max(CA_{test}) Results from the SNR Screening Method for TDNN

K	J	I	max(CA_{test})	CA_{train}
2	1	5	87.20%	84.80%
	2	6	86.13%	92.27%
	3	9	85.33%	98.40%
	4	7	84.27%	97.07%
	8	11	83.73%	100.00%
	12	14	83.20%	100.00%
3	1	20	57.60%	48.80%
	2	7	60.80%	72.53%
	3	6	58.67%	73.33%
	4	24	57.33%	80.27%
	8	13	56.27%	93.33%
	12	10	55.47%	91.47%

For each TDNN architecture, Table 34 lists the maximum $\overline{CA}_{test}$ attained from applying the SNR screening method thirty times. In addition, Table 34 lists the number of salient features and that produced the maximum $\overline{CA}_{test}$ in addition to the associated $\overline{CA}_{train}$. The results are given for the $K = 2$ class workload problem (VFR/IFR) and the $K = 3$ class workload problem (low/medium/high).

For each TDNN architecture, Table 35 lists the maximum CA_{test} denoted as max(CA_{test}) attained during thirty replications of the SNR screening method. In addition, Table 35 lists the number of salient features I that produced the maximum CA_{test} in addition to the associated CA_{train}.

For each TDNN architecture, Table 36 lists the parsimonious set of salient features and their rankings resulting from the SNR screening method that produced the maximum CA_{test} in Table 35. In Table 36, EB is for number of eye blinks, HR is for heart rate, and RR is for respiration rate. Table 37 summarizes the results attained after

the TDNN was retrained thirty times using only the salient features without the injected noise feature. For both workload classification problems, there were 30 TDNNs trained using the architecture with the set of salient features that produced the maximum $\overline{CA}_{test}$ and 30 TDNNs trained using the architecture with the set of salient features that produced the maximum CA_{test}.

8.4.4.2.1 Visual Flight Rules (VFR) / Instrument Flight Rules (IFR) Classification Problem

For the VFR/IFR classification problem, the $11/4/2$ TDNN architecture produced the maximum $\overline{CA}_{test} = 80.12\%$ as highlighted in Table 34. Of the thirty replications of the SNR screening method as applied to $I/4/2$ TDNN architectures, the best replication produced $CA_{test} = 84.27\%$ with $I = 7$ salient input features as shown in Table 35. But, the replication that produced the overall highest $CA_{test} = 87.20\%$ was a $5/1/2$ TDNN as highlighted in Table 35.

For each architecture, Table 36 lists the parsimonious set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the $7/4/2$ TDNN architecture, the salient features in order were:

1. HR(t-2)
2. HR(t-7)
3. EB9(t-4)
4. EB(t-8)
5. EB(t)
6. EB(t-6)
7. HR(t-6)

246

Table 36. Feature Ranking Results from SNR Screening Method for TDNN

K	J = 1	J = 2	J = 3	J = 4	J = 8	J = 12
2	1. EB(t-8)	1. HR(t-8)	1. HR(t-2)	1. HR(t-2)	1. HR(t-1)	1. HR(t-6)
	2. EB(t)	2. EB(t-8)	2. EB(t-4)	2. HR(t-7)	2. HR(t-6)	2. EB(t-8)
	3. EB(t-2)	3. EB(t-4)	3. HR(t-1)	3. EB(t-4)	3. EB(t-4)	3. HR(t-2)
	4. EB(t-4)	4. EB(t)	4. EB(t-8)	4. EB(t-8)	4. HR(t-4)	4. RR(t-7)
	5. EB(t-6)	5. EB(t-2)	5. HR(t-6)	5. EB(t)	5. EB(t-6)	5. EB(t)
		6. EB(t-6)	6. EB(t-6)	6. EB(t-6)	6. EB(t-8)	6. EB(t-6)
		7. RR(t)	7. EB(t)	7. HR(t-6)	7. HR(t-2)	7. HR(t-1)
			8. HR(t-8)		8. RR(t-7)	8. EB(t-2)
			9. EB(t-2)		9. EB(t)	9. HR(t-7)
					10. HR(t-8)	10. EB(t-4)
					11. HR(t-5)	11. HR(t-5)
						12. HR(t-8)
						13. EB(t-7)
						14. EB(t-5)
3	1. EB(t-1)	1. EB(t)	1. HR(t-1)	1. HR(t-6)	1. HR(t-8)	1. HR(t-6)
	2. EB(t-8)	2. HR(t-2)	2. EB(t-2)	2. HR(t-8)	2. HR(t-2)	2. EB(t-2)
	3. EB(t-5)	3. EB(t-8)	3. EB(t)	3. HR(t)	3. EB(t-6)	3. EB(t)
	4. EB(t)	4. HR(t-4)	4. EB(t-4)	4. HR(t-3)	4. EB(t-8)	4. HR(t-1)
	5. RR(t-8)	5. EB(t-2)	5. EB(t-8)	5. EB(t-2)	5. HR(t-6)	5. EB(t-6)
	6. HR(t-1)	6. EB(t-6)	6. HR(t-3)	6. HR(t-4)	6. RR(t-8)	6. HR(t-4)
	7. EB(t-6)	7. HR(t-1)		7. RR(t-6)	7. EB(t)	7. HR(t-7)
	8. RR(t-1)			8. EB(t-4)	8. HR(t-4)	8. EB(t-4)
	9. EB(t-2)			9. HR(t-2)	9. RR(t-2)	9. EB(t-1)
	10. HR(t)			10. RR(t-4)	10. EB(t-4)	10. RR(t-8)
	11. HR(t-8)			11. RR(t-8)	11. EB(t-2)	
	12. RR(t-6)			12. EB(t-6)	12. RR(t-4)	
	13. RR(t-3)			13. RR(t-2)	13. HR(t-5)	
	14. HR(t-6)			14. RR(t)		
	15. HR(t-3)			15. EB(t-5)		
	16. HR(t-7)			16. EB(t)		
	17. HR(t-5)			17. HR(t-5)		
	18. EB(t-3)			18. RR(t-5)		
	19. EB(t-4)			19. EB(t-8)		
	20. RR(t-4)			20. RR(t-1)		
				21. HR(t-1)		
				22. HR(t-7)		
				23. EB(t-1)		
				24. RR(t-7)		

For the 5/1/2 TDNN architecture, the salient features in order were:

1. EB(t-8)
2. EB(t)
3. EB(t-2)
4. EB(t-4)
5. EB(t-6)

Thirty 7/4/2 TDNNs with the salient features listed in Table 36 were trained without noise. In addition, thirty 5/1/2 TDNNs with the salient features listed in Table 36 were trained without noise. The results are shown in Table 37. For the VFR/IFR classification problem, the maximum $\overline{CA}_{test} = 87.57\%$ and the maximum $CA_{test} = 88.53\%$ was attained using a 5/1/2 TDNN with number of eye blinks at time window t, $t-2$, $t-4$, $t-6$, and $t-8$. Figure 64 provides plots of the actual and desired outputs for the TDNN listed in Table 37 for the VFR/IFR classification problem.

The parsimonious set of salient features selected provides an interested result in that no *redundant* lags of the number of eye blinks were selected. Each input feature was averaged over a 10-second moving window with 50% overlap. The SNR screening method selected the number of eye blinks at time window t, $t-2$, $t-4$, $t-6$, and $t-8$ which provided no overlapping information. Figure 65 provides further elaboration on this point. Figure 65 pictorially shows the 10-second moving windows with 50% overlap for the number of eye blinks at time window t, $t-1$, $t-2$, ..., $t-8$.

Table 37. Best Results without Noise for TDNN

K	J	I	$\overline{CA}_{test}$	$\overline{CA}_{train}$	$\max(CA_{test})$	CA_{train}
2	1	5	87.57%	86.64%	88.53%	86.93%
	4	7	79.87%	98.42%	82.67%	98.67%
3	2	7	54.42%	71.15%	56.00%	72.53%
	3	6	53.66%	71.94%	58.40%	73.07%

Figure 64. Actual and Desired Plots for Two-Class TDNN
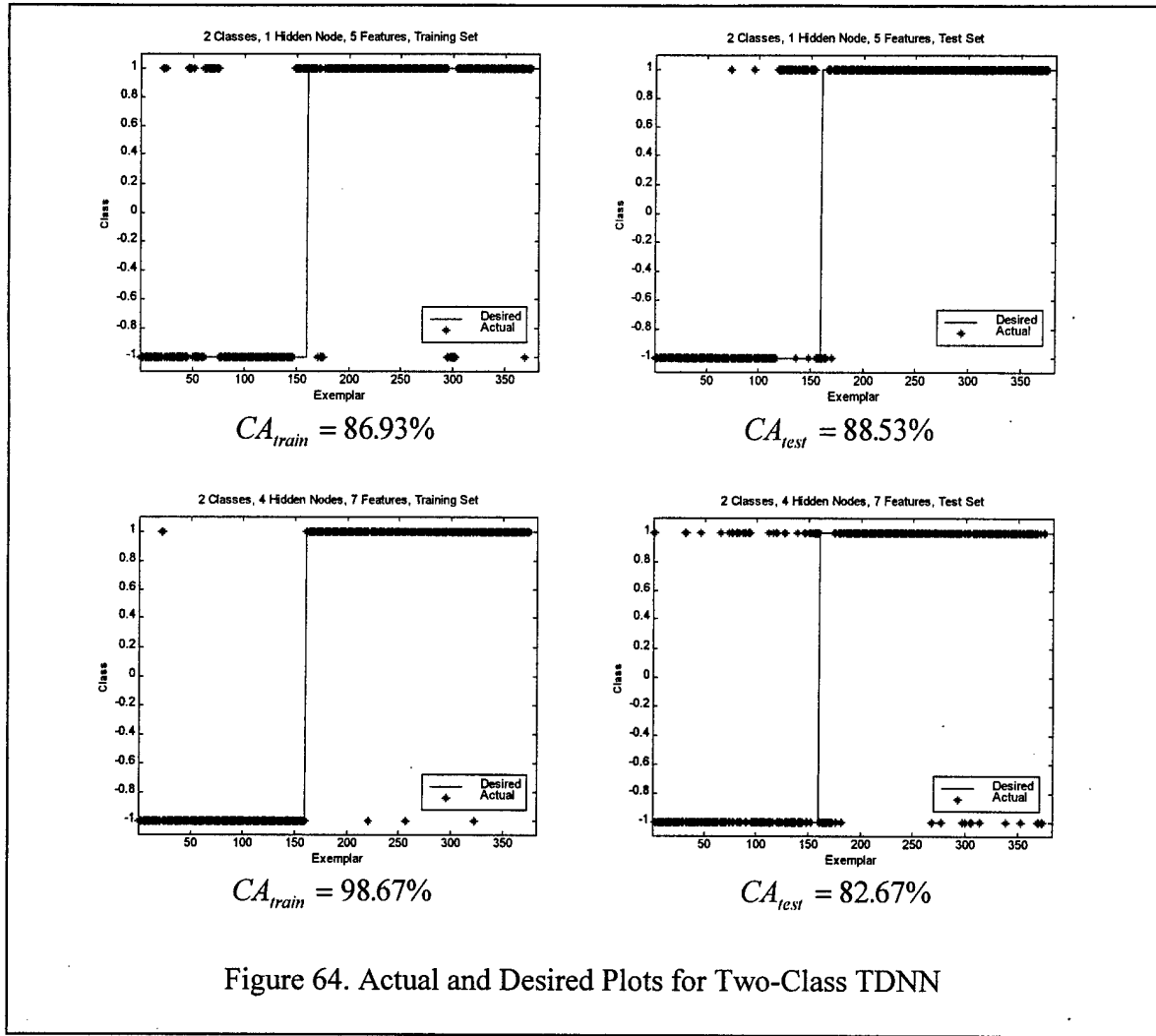
8.4.4.2.2 Low/Medium/High Workload Classification Problem

For the low/medium/high workload classification problem, the 15 / 3 / 3 TDNN architecture produced the maximum $\overline{CA_{test}} = 50.39\%$ as highlighted in Table 34. Of the thirty replications of the SNR screening method as applied to I / 3 / 3 TDNNs, the best replication produced $CA_{test} = 58.67\%$ with $I = 6$ salient input features as shown in Table 35. But, the replication that produced the overall highest $CA_{test} = 60.80\%$ was a 7 / 2 / 3 TDNN as highlighted in Table 35.
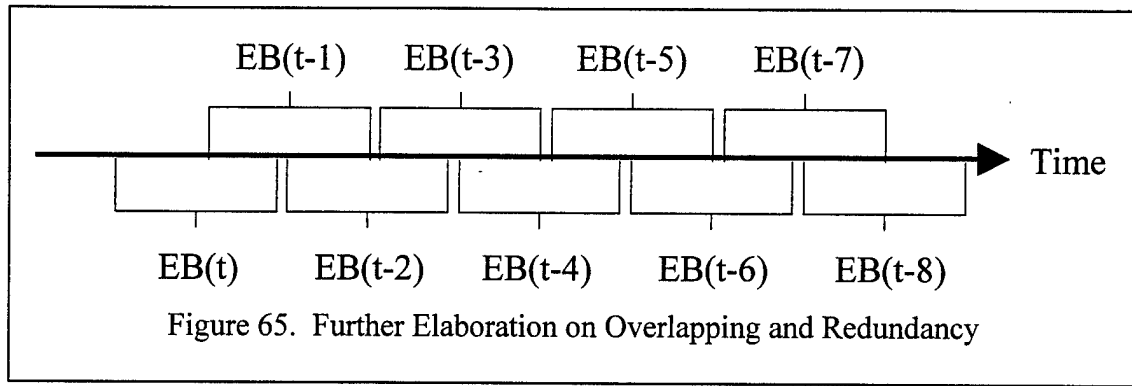
EB(t-1)   EB(t-3)   EB(t-5)   EB(t-7)

Time

EB(t)   EB(t-2)   EB(t-4)   EB(t-6)   EB(t-8)

Figure 65. Further Elaboration on Overlapping and Redundancy

For each architecture, Table 36 lists the parsimonious set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the 6/3/3 TDNN architecture, the salient features in order were:

1. HR(t-1)
2. EB(t-2)
3. EB(t)
4. EB(t-4)
5. EB(t-8)
6. HR(t-3)

as highlighted in Table 36. For the 7/2/3 TDNN architecture, the salient features in order were:

1. EB(t)
2. HR(t-2)
3. EB(t-8)
4. HR(t-4)
5. EB(t-2)
6. EB(t-6)
7. HR(t-1)

as highlighted in Table 36.

Thirty 6/3/3 TDNNs with the salient features listed in Table 36 were trained without noise. In addition, thirty 7/2/3 TDNNs with the salient features listed in Table 36 were trained without noise. The results are shown in Table 37. For the three-class pilot workload problem, the maximum $\overline{CA_{test}} = 54.42\%$ was attained using a 7/2/3

3 Classes, 2 Hidden Nodes, 7 Features, Training Set — $CA_{train} = 72.53\%$

3 Classes, 2 Hidden Nodes, 7 Features, Test Set — $CA_{test} = 56.00\%$

3 Classes, 3 Hidden Nodes, 6 Features, Training Set — $CA_{train} = 73.07\%$

3 Classes, 3 Hidden Nodes, 6 Features, Test Set — $CA_{test} = 58.40\%$

Figure 66. Actual and Desired Plots for Three-Class TDNN
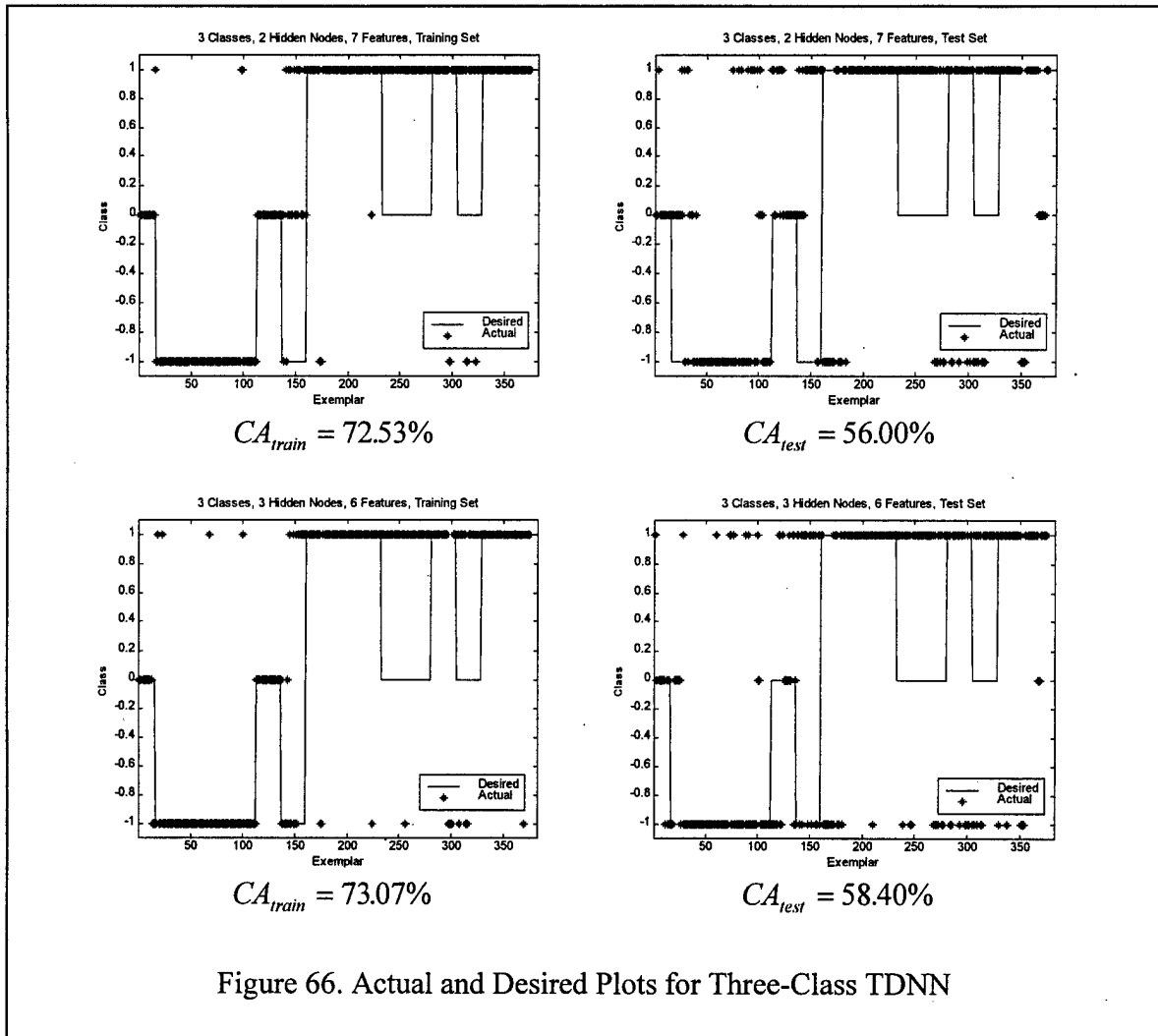
TDNN. The maximum $CA_{test} = 58.40\%$ was attained using a 6/3/3 TDNN. Figure 66 provides plots of the actual and desired outputs for the feedforward MLP ANNs listed in Table 37 for the three-class pilot workload problem.

8.4.4.3 Elman Recurrent Neural Network (RNN) Experimental Design

Table 38 summarizes the average classification accuracy denoted as $\overline{CA}$ from applying the spatial-temporal feature screening method thirty times to an Elman RNN.

251

Table 38. $\overline{CA}$ Results from Spatial-Temporal Screening Method for Elman RNN

K	J	$\overline{CA}_{train}(3)$	$\overline{CA}_{test}(3)$	$\overline{CA}_{test}(2)$	$\overline{CA}_{test}(1)$	$\overline{CA}_{test}(N)$
2	1	69.51%	67.98%	69.56%	68.75%	51.47%
	2	88.76%	82.73%	84.15%	86.00%	61.50%
	3	94.39%	86.86%	88.10%	83.57%	62.44%
	4	96.39%	89.16%	88.05%	83.32%	59.63%
3	1	45.20%	41.43%	41.35%	38.99%	35.22%
	2	58.58%	48.88%	51.64%	44.97%	37.48%
	3	59.16%	48.51%	49.70%	39.03%	40.06%
	4	66.69%	53.46%	55.77%	45.24%	41.25%

The results are given for the $K = 2$ class workload problem (VFR/IFR) and the $K = 3$ class workload problem (low/medium/high).

For each Elman RNN architecture, Table 39 lists the maximum CA_{test} denoted as $\max(CA_{test})$ attained from applying the spatial-temporal feature screening method thirty times. In addition, Table 39 lists the number of salient features I that produced the maximum CA_{test} in addition to the associated CA_{train}.

For each Elman RNN architecture, Table 40 lists the parsimonious set of salient features and their rankings from the spatial-temporal feature screening method that produced the maximum CA_{test} listed in Table 39. Table 41 summarizes the results attained after the Elman RNN was retrained thirty times using only the salient features without the injected noise feature. For the two-class problem (VFR/IFR), there were thirty Elman RNNs trained using the architecture with the set of salient features that produced the maximum $\overline{CA}_{test}$ and thirty Elman RNNs trained using the architecture with the set of salient features that produced the maximum CA_{test}. For the three-class problem (low/medium/high), the same architecture with the same set of salient features produced the maximum $\overline{CA}_{test}$ and the maximum CA_{test}. So for the three-class problem

252

Table 39. $\max(CA_{test})$ Results from Spatial-Temporal Feature Screening Method for Elman RNN

K	J	I	$\max(CA_{test})$	CA_{train}
2	1	2	97.33%	93.33%
	2	1	98.13%	96.00%
	3	1	99.20%	96.27%
	4	2	99.20%	90.93%
3	1	1	67.47%	67.73%
	2	2	68.00%	63.73%
	3	3	64.27%	73.33%
	4	2	70.13%	77.60%

(low/medium/high), there were thirty Elman RNNs trained using the architecture with the set of salient features that produced both the maximum $\overline{CA_{test}}$ and the maximum CA_{test}. Because the Elman RNN has a high tendency to converge to local minima via backpropagation, the minimum CA_{test} and the associated CA_{train}. This high likelihood of training to a local minima negatively effected the $\overline{CA_{test}}$ and $\overline{CA_{train}}$ results for the Elman RNN and should be taken into consideration when comparing the results between the feedforward MLP ANN, the TDNN, and the Elman RNN.

8.4.4.3.1 Visual Flight Rules (VFR) / Instrument Flight Rules (IFR) Classification Problem

For the VFR/IFR classification problem, the $3+4/4/2$ Elman RNN architecture produced the maximum $\overline{CA_{test}} = 89.16\%$ as highlighted in Table 38. Of the thirty replications of the spatial-temporal feature screening method as applied to $I+4/4/2$ Elman RNNs, the best replication was a $2+4/4/2$ Elman RNN architecture that produced $CA_{test} = 99.20\%$ with an associated $CA_{train} = 90.93\%$ as shown in Table 39.

Table 40. Results from Spatial-Temporal Feature Screening Method for Elman RNN

K	J = 1	J = 2	J = 3	J = 4
2	1. EB	1. EB	1. EB	1. EB
	2. RR			2. HR
3	1. EB	1. HR	1. HR	1. HR
		2. EB	2. EB	2. EB
			3. RR	

Table 41. Best Results without Noise for Elman RNN

K	J	I	$\overline{CA_{test}}$	$\overline{CA_{train}}$	$\max(CA_{test})$	CA_{train}	$\min(CA_{test})$	CA_{train}
2	3	1	84.83%	87.49%	94.40%	97.60%	42.67%	42.67%
	4	2	88.10%	95.75%	99.47%	100.00%	57.33%	57.33%
3	1	3	58.25%	69.04%	69.33%	77.60%	46.13%	56.80%

The best replication of the $1+3/3/2$ Elman RNN architecture also produced $CA_{test} = 99.20\%$ as highlighted in Table 39 but with a slightly better $CA_{train} = 96.27\%$ and with one less feature. Table 40 lists the set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the $2+4/4/2$ Elman RNN architecture, the salient features in order were:

1. Number of eye blinks
2. Heart rate

as highlighted in Table 40. For the $1+3/3/2$ Elman RNN architecture, the only salient feature was number of eye blinks.

Thirty $2+4/4/2$ Elman RNNs were trained with number of eye blinks and heart rate as inputs with no noise. In addition, thirty $1+3/3/2$ Elman RNNs were trained with number of eye blinks as the only input with no noise. The results are shown in Table 41. For the VFR/IFR classification problem, the maximum $\overline{CA_{test}} = 88.10\%$ and the maximum $CA_{test} = 99.47\%$ were attained using a $2+4/4/2$ Elman RNNs were

254

2 Classes, 3 Hidden Nodes, 1 Feature (Number of Eyeblinks), Training Set

2 Classes, 3 Hidden Nodes, 1 Feature (Number of Eyeblinks), Test Set

$CA_{train} = 97.60\%$

$CA_{test} = 94.40\%$

2 Classes, 4 Hidden Nodes, 3 Features, Training Set

2 Classes, 4 Hidden Nodes, 3 Features, Test Set

$CA_{train} = 100.00\%$

$CA_{test} = 99.47\%$
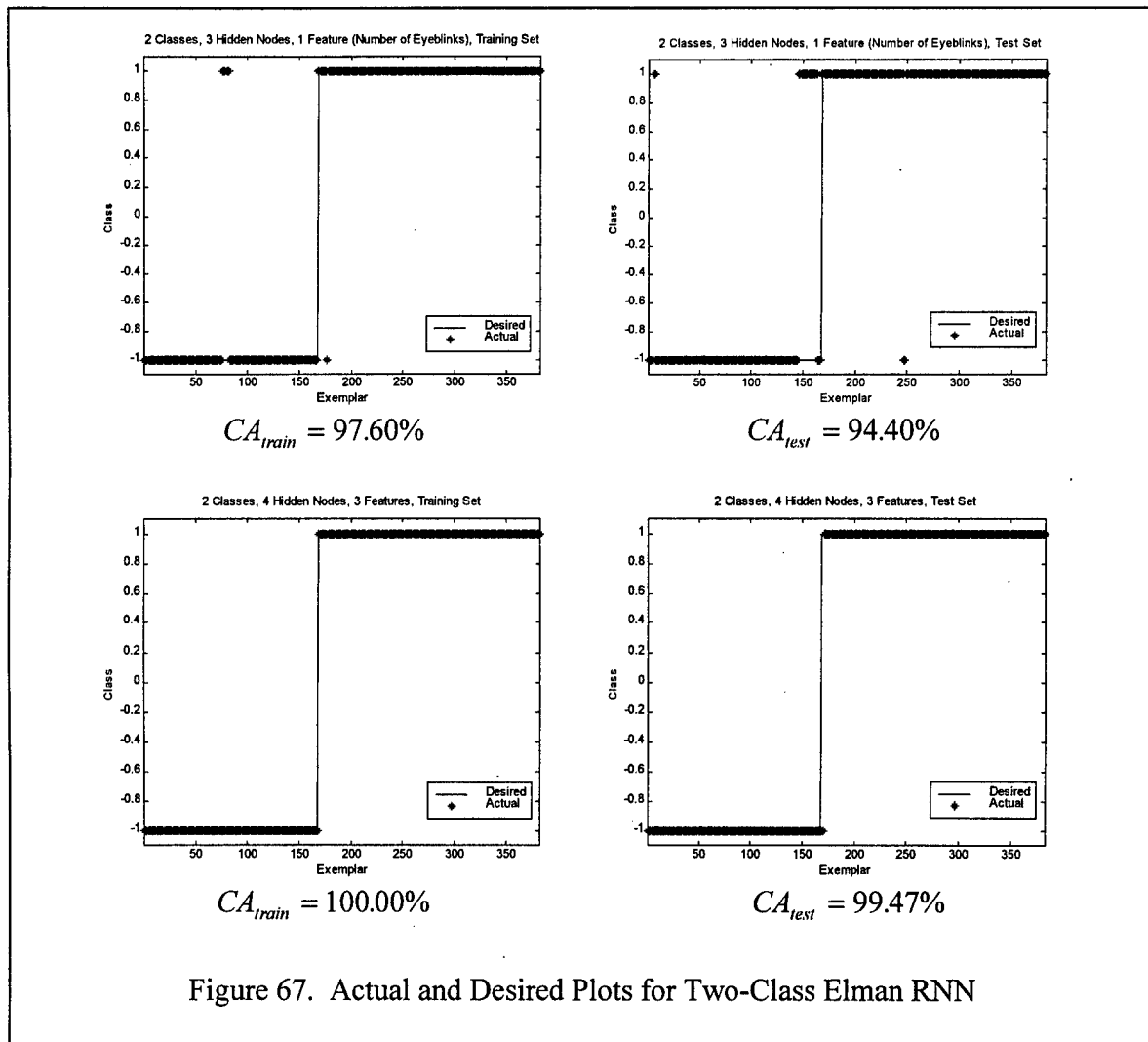
Figure 67. Actual and Desired Plots for Two-Class Elman RNN

trained with number of eye blinks and respiration rate as inputs. Figure 67 provides plots of the actual and desired output for the Elman RNNs in Table 41 for the VFR/IFR classification problem.

8.4.4.3.2 Low/Medium/High Workload Classification Problem

For the VFR/IFR classification problem, the $2 + 4 / 4 / 3$ Elman RNN architecture produced the maximum $\overline{CA}_{test} = 55.77\%$ as highlighted in Table 38. Of the thirty replications of the spatial-temporal feature screening method as applied to $I + 4 / 4 / 3$

255

Elman RNNs, the best replication was a $2+4/4/3$ Elman RNN architecture that produced $CA_{test} = 70.13\%$ as shown in Table 39.

For each architecture, Table 40 lists the set of salient features and their rankings for the replication that produced the maximum CA_{test}. For the $2+4/4/3$ Elman RNN architecture, the salient features in order were:

1. Number of eye blinks
2. Heart rate

as highlighted in Table 40. Thirty $2+4/4/3$ Elman RNNs were trained with number of eye blinks and heart rate as inputs with no noise. The results are shown in Table 41. For the low/medium/high workload classification problem, the maximum $\overline{CA}_{test} = 58.25\%$ and the maximum $CA_{test} = 69.33\%$. Figure 63 provides plots of the actual and desired outputs for the Elman RNN listed in Table 41 for the three-class pilot workload problem.

8.4.5 Conclusions

By using a TDNN instead of a feedforward MLP ANN, the maximum $\overline{CA}_{test}$ was

$$CA_{train} = 77.60\%$$ $$CA_{test} = 69.33\%$$

Figure 68. Actual and Desired Plots for Three-Class Elman RNN

256
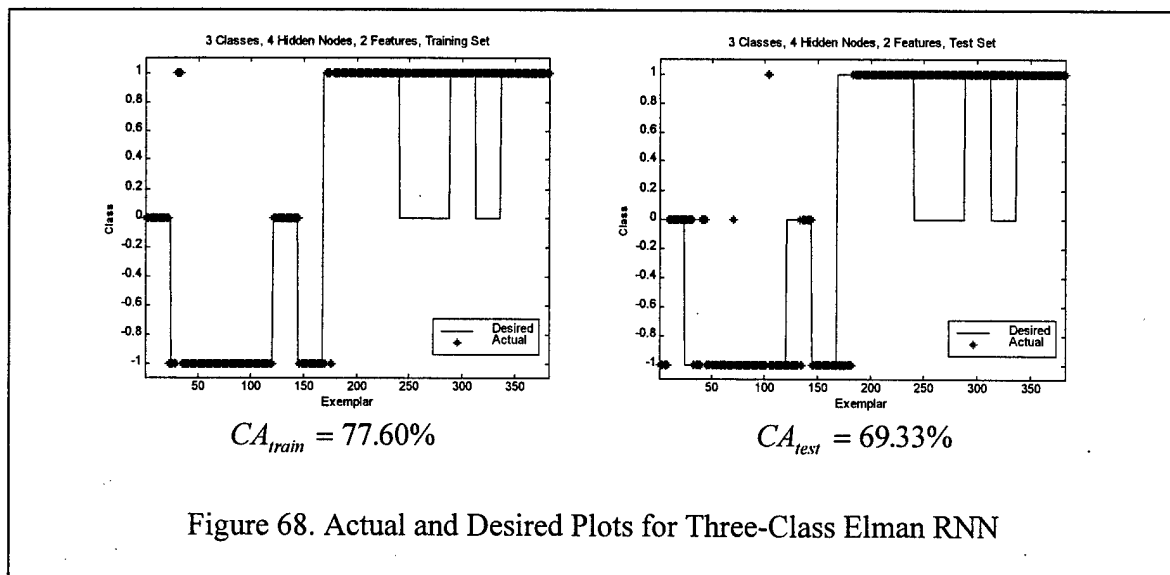
improved by 17.34% and the maximum CA_{test} was improved by 18.30% for the VFR/IFR classification problem. The maximum $\overline{CA_{test}}$ was improved by 11.04% and the maximum CA_{test} was improved by 13.49% by using a TDNN for the low/medium/high workload classification problem.

By using an Elman RNN instead of a feedforward MLP ANN, the maximum CA_{test} was improved by 29.24% for the VFR/IFR classification problem. The maximum CA_{test} was improved by 24.42% by using a Elman RNN for the low/medium/high workload classification problem.

Table 42 provides the average CPU time in minutes required to perform the feature screening method used. For the feedforward MLP ANN and the TDNN, the SNR screening method was used. For the Elman RNN, the spatial-temporal screening method was used. Though the spatial-temporal feature screening method and the Elman RNNs produced the best results, those results came at the cost of CPU. A trade-off exists between classification accuracy performance and CPU.

Table 42. Average CPU Time in Minutes to Perform Feature Screening Method

K	ANN	$J = 1$	$J = 2$	$J = 3$	$J = 4$	$J = 8$	$J = 12$
2	MLP	0.89	1.09	1.18	1.19	1.47	1.76
	TDNN	5.30	7.36	7.93	8.34	11.02	14.04
	Elman RNN	39.07	37.43	40.18	52.61		
3	MLP	1.30	1.16	1.26	1.29	1.61	1.89
	TDNN	7.08	9.55	10.34	11.43	15.06	19.05
	Elman RNN	40.30	45.53	66.19	94.08		

9.1   Introduction

This chapter contains a methodology for determining the *memory capacity* of an Elman RNN [48]. The memory capacity of an Elman RNN is defined in terms of the number of *unfolded layers* containing salient input and context nodes. Researchers are interested in determining how far back in time RNNs *remember*. In other words, how far back in time do the input and context nodes effect the current output of an Elman RNN? This chapter provides mathematical derivations for determining the memory capacity of an Elman RNN performing a wave amplitude detection problem, a well known nonlinear process, by *unfolding through time*. The proposed method is based on partial derivatives calculated over time. The approach calculates the partial derivatives over time of the output of a trained Elman RNN relative to the input. In addition, the partial derivatives over time of the output relative to the context nodes are calculated. The partial derivatives over time relative to the input and context nodes are statistically compared to that of an injected noise feature. This injected noise feature provides a baseline for determining the unfolded layer at which the input and context nodes provide no more information than noise to the Elman RNN.

9.2   Data

A wave amplitude detection problem is shown in Figure 69. A signal $x(t)$ that varies between two amplitudes is inputted to the Elman RNN.
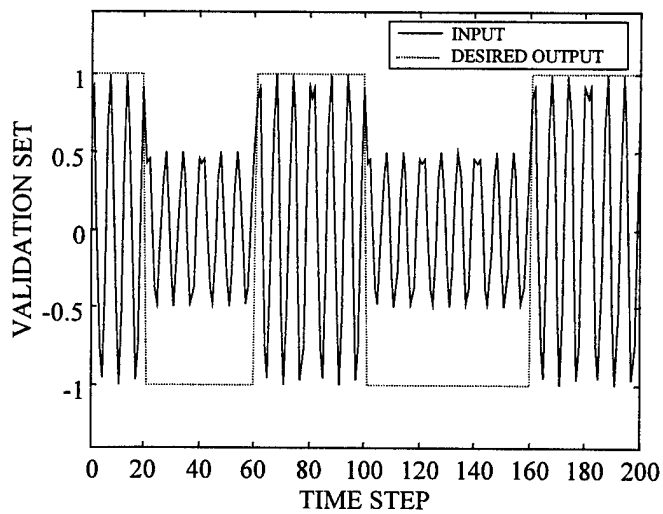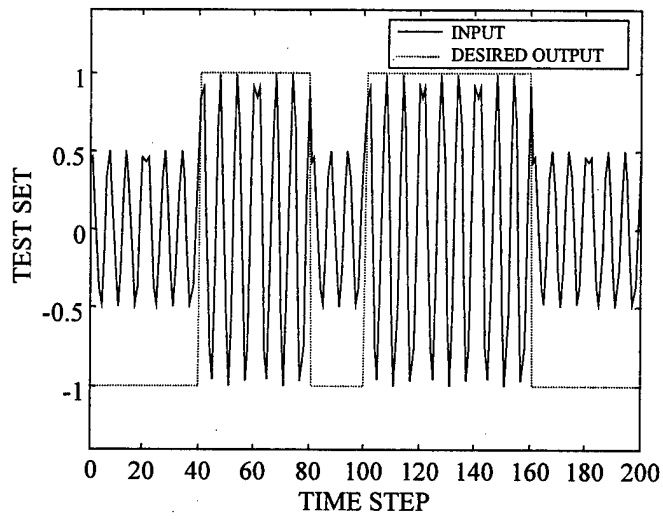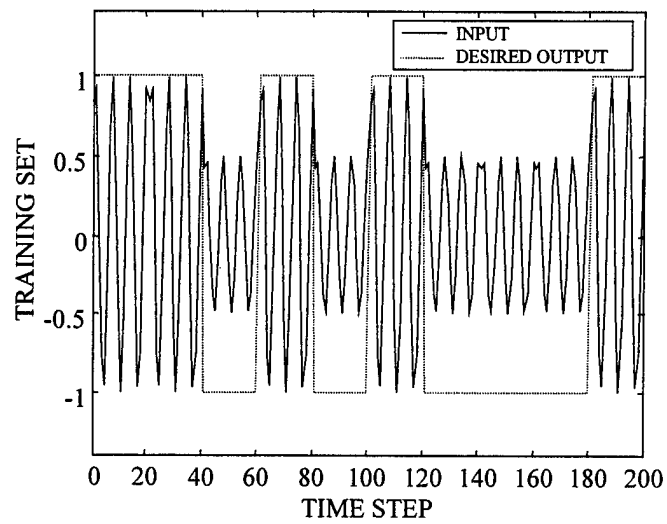
Figure 69. Wave Amplitude Detection Problem

The desired output of the Elman RNN at time t denoted as $d(t)$ is a nonlinear metafunction of the input signal $x(t)$ and is defined as:

$$d(t) = \begin{cases} -1.0 & \text{if } amplitude(x(t)) = 0.5 \\ +1.0 & \text{if } amplitude(x(t)) = 1.0 \end{cases}$$
(171)

9.3 Methodology

An Elman RNN with a $2+2/2/1$ architecture as shown in Figure 70 is utilized. The input layer contains:

- One injected uniform (-1.0, 1.0) noise node [11, 12, 147] at time t denoted as $noise(t)$.
- One input node at time t denoted as $x(t)$.
- Two context nodes at time $t-1$ denoted as $y_{j^0}(t-1)$ for $j^0 = 1,2$.
- One bias node.

The hidden layer contains:

- Two hidden nodes at time t denoted as $y_j(t)$ for $j = 1,2$.
- One bias node.

The output layer contains:

- One output node at time t denoted as $z(t)$.

All hidden and context nodes are activated with the hyperbolic tangent nonlinear transfer function in Equation 6. The linear transfer function with slope $= 1$ in Equation 8 activates the output node.

9.3.1 Partial Derivative-Based Saliency Measure in Elman Recurrent Neural Networks (RNN)

For the Elman RNN as depicted in Figure 70, the partial derivative-based saliency

260

Figure 70. $2 + 2 / 2 / 1$ Elman RNN
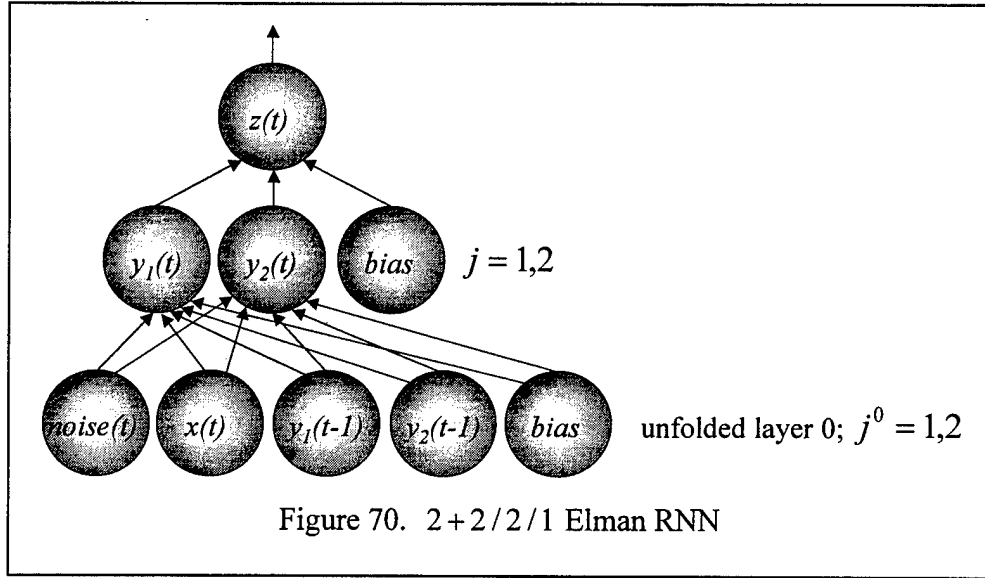
measure for x is calculated using the training set exemplars following Equation 66 as:

$$\Gamma_x = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \frac{\partial z(t, \mathbf{W})}{\partial x(t)} \right| \tag{172}$$

where Γ_x is the partial derivative-based saliency measure for x, T is the number of time

steps, $z(t, \mathbf{W})$ is the activation of the output node at time t with the trained weight matrix

$\mathbf{W}$, and $x(t)$ is the input at time t. More specifically:

$$\Gamma_x = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \dot{f}_1^2 \left(a_1^2(t, \mathbf{W}) \right) \cdot \sum_{j=1}^{2} w_j^2 \cdot \dot{f}_j^1 \left(a_j^1(t, \mathbf{W}) \right) \cdot w_{x,j}^1 \right| \tag{173}$$

Equation 173 becomes:

$$\Gamma_x = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - \left(y_j(t, \mathbf{W}) \right)^2 \right) \cdot w_{x,j}^1 \right| \tag{174}$$

since $\dot{f}(a) = 1$ for linear transfer functions and $\dot{f}(a) = 1 - (f(a))^2$ for hyperbolic tangent

nonlinear transfer functions. The partial derivative-based saliency is calculated in the

same manner for the noise node as:

261

$$\Gamma_{noise} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - \left(y_j(t, \mathbf{W})\right)^2\right) \cdot w_{noise,j}^1 \right| \qquad (175)$$

where Γ_{noise} is the partial derivative-based saliency measure for the noise node and $w_{noise,j}^1$ is the first layer weight connecting the noise node to hidden node y_j. The partial derivative-based saliency is calculated in the same manner for context node y_{j^0} on the input layer as:

$$\Gamma_{j^0} = \frac{1}{T} \cdot \sum_{t=1}^{T} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - \left(y_j(t, \mathbf{W})\right)^2\right) \cdot w_{j^0,j}^1 \right| \quad \text{for } j^0 = 1,2 \qquad (176)$$

where Γ_{j^0} is the partial derivative-based saliency measure for context node y_{j^0} and $w_{j^0,j}^1$ is the first layer weight connecting context node y_{j^0} to hidden node y_j.

9.3.2 Partial Derivative-Based Saliency Measure Over Time in Elman Recurrent Neural Networks (RNN)

An Elman RNN can be viewed as a feedforward MLP ANN which has been folded back onto itself in time. The Elman RNN can be *unfolded through time* as depicted in Figure 71. Unfolding each layer of the Elman RNN allows us to visualize the input and hidden layers that affect $z(t)$. This unfolding of an Elman RNN shows the effect of the temporal feedback in a spatial representation. The partial derivative-based saliency measure can be calculated for each unfolding of the Elman RNN by extending the equations in Section 9.3.1. For the unfolded layer 1, the partial derivative-based saliency measure for x is calculated as:

$$\Gamma_x^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \frac{\partial z(t, \mathbf{W})}{\partial x(t-1)} \right| \qquad (177)$$
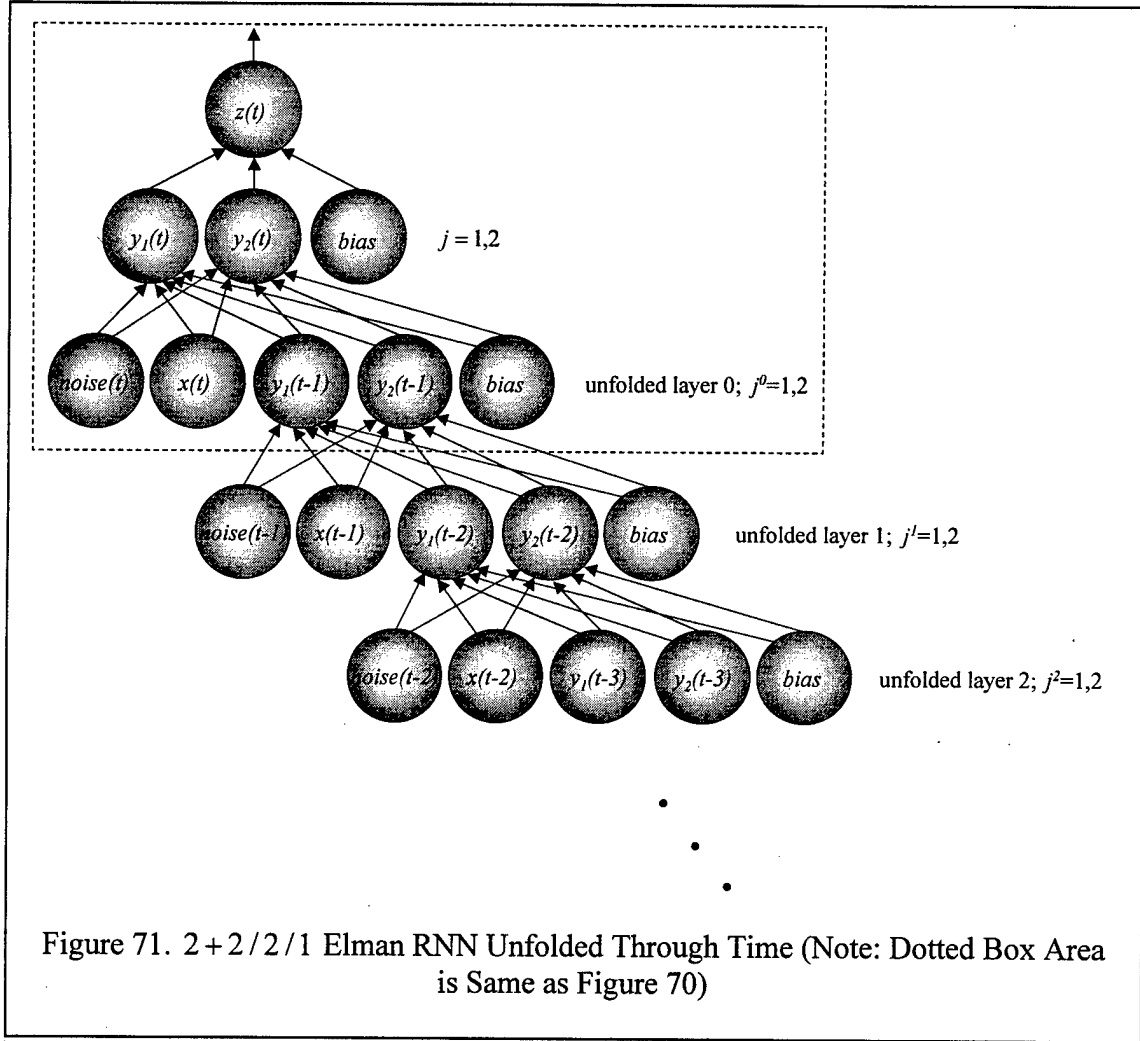
Figure 71. $2 + 2/2/1$ Elman RNN Unfolded Through Time (Note: Dotted Box Area is Same as Figure 70)

where Γ_x^1 is the partial derivative-based saliency measure for x on unfolded layer 1. More specifically,

$$\Gamma_x^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - \left(y_j(t, \mathbf{W})\right)^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - \left(y_{j^0}(t-1, \mathbf{W})\right)^2\right) \cdot w_{x,j^0}^1 \right| \quad (178)$$

The partial derivative-based saliency is calculated in the same manner for the noise node on unfolded layer 1 as:

$$\Gamma_{noise}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - \left(y_j(t, \mathbf{W})\right)^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - \left(y_{j^0}(t-1, \mathbf{W})\right)^2\right) \cdot w_{noise,j^0}^1 \right| \quad (179)$$

263

where Γ_{noise}^1 is the partial derivative-based saliency measure for the noise node on unfolded layer 1. The partial derivative-based saliency is calculated for the two context nodes y_{j^1} on unfolded layer 1 as:

$$\Gamma_{y_{j^1}}^1 = \frac{1}{T-1} \cdot \sum_{t=1}^{T-1} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - (y_j(t, \mathbf{W}))^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - (y_{j^0}(t-1, \mathbf{W}))^2\right) \cdot w_{j^1,j^0}^1 \right| \quad \text{for } j^1 = 1, 2 \quad (180)$$

where $\Gamma_{y_{j^1}}^1$ is the partial derivative-based saliency measure for context node y_{j^1} on unfolded layer 1.

For the unfolded layer 2, the partial derivative-based saliency measure for x is calculated as:

$$\Gamma_x^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \frac{\partial z(t, \mathbf{W})}{\partial x(t-2)} \right| \quad (181)$$

where Γ_x^2 is the partial derivative-based saliency measure for x on unfolded layer 2. More specifically,

$$\Gamma_x^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - (y_j(t, \mathbf{W}))^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - (y_{j^0}(t-1, \mathbf{W}))^2\right) \cdot \sum_{j^1=1}^{2} w_{j^1,j^0}^1 \cdot \left(1 - (y_{j^1}(t-2, \mathbf{W}))^2\right) \cdot w_{x,j^1}^1 \right| \quad (182)$$

The partial derivative-based saliency is calculated in the same manner for the noise node on unfolded layer 2 as:

$$\Gamma_{noise}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - (y_j(t, \mathbf{W}))^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - (y_{j^0}(t-1, \mathbf{W}))^2\right) \cdot \sum_{j^1=1}^{2} w_{j^1,j^0}^1 \cdot \left(1 - (y_{j^1}(t-2, \mathbf{W}))^2\right) \cdot w_{noise,j^1}^1 \right| \quad (183)$$

where Γ_{noise}^2 is the partial derivative-based saliency measure for the noise node on unfolded layer 2. The partial derivative-based saliency is calculated for the two context nodes y_{j^2} on unfolded layer 2 as:

264

$$\Gamma_{y_{j^2}}^2 = \frac{1}{T-2} \cdot \sum_{t=1}^{T-2} \left| \sum_{j=1}^{2} w_j^2 \cdot \left(1 - (y_j(t, \mathbf{W}))^2\right) \cdot \sum_{j^0=1}^{2} w_{j^0,j}^1 \cdot \left(1 - (y_{j^0}(t-1, \mathbf{W}))^2\right) \cdot \sum_{j^1=1}^{2} w_{j^1,j^0}^1 \cdot \left(1 - (y_{j^1}(t-2, \mathbf{W}))^2\right) \cdot w_{j^2,j^1}^1 \right| \quad \text{for } j^2 = 1,2 \quad (184)$$

where $\Gamma_{y_{j^2}}^1$ is the partial derivative-based saliency measure for context node y_{j^2} on

unfolded layer 2. Following this logic, the partial derivatives can be calculated as far

back in time as desired.

9.3.3 Training

All weights and biases were initialized using the Nguyen-Widrow method [102].

The initial context nodes were set to 0.0. All Elman RNNs were trained using the *Matlab*

Neural Network Toolbox implementation of gradient descent backpropagation with

momentum and an adaptive learning rate following Equation 50 with $m_c = 0.90$ and

initially, $\eta = 0.01$. The momentum and adaptive learning rate were implemented as

described in Section 2.4.11. After training for 500 epochs, the weights for the epoch that

produced the minimum SSE_{test} is kept thus preventing memorization of the training set.

The validation set is used to validate the set of weights selected during the training-test

phase.

9.4 Results

Fifty-two Elman RNNs were trained in order to get 30 sufficiently trained Elman

RNNs. Thirty Elman RNNs were sufficiently trained so Central Limit tendencies can be

exploited [88]. An Elman RNN is sufficiently trained if $CA_{train} > 90\%$, $CA_{test} > 90\%$,

and $CA_{valid} > 90\%$. An output $z(t)$ is considered to correctly classify the input signal

$x(t)$ when:

265

$$z(t) = \begin{cases} \geq 0.0 & \text{and } d(t) = 1.0 \\ < 0.0 & \text{and } d(t) = -1.0 \end{cases} \tag{185}$$

Of the 52 Elman RNNs that were trained, only 58% provided sufficient classification accuracies. Table 43 summarizes the classification accuracies where $\overline{CA}$ is the average classification accuracy and $S_{\overline{CA}}$ is the standard deviation of the average classification accuracy. It appears from these results that the specific Elman RNN applied to this problem has a high likelihood of training to a local minimum. Techniques utilizing simulated annealing may correct for the Elman RNN's apparent high probability of local minima.

The partial derivatives up to eight unfolded layers for the 30 sufficiently trained Elman RNNs was computed. Table 44 summarizes the partial derivatives where $\overline{\Gamma}$ is the average partial derivative and $S_{\overline{\Gamma}}$ is the standard deviation of the average partial derivative. Instead of taking the average over 30 sufficiently trained Elman RNNs for each context node, the average of the context node that results in the maximum and minimum partial derivative was taken.

As in the case of unfolded layer 1:

$$\overline{\Gamma}^1_{y_{max}} = \frac{\sum_{n=1}^{N} \max\left\{\Gamma^1_{y_1}(n), \Gamma^1_{y_2}(n)\right\}}{N} \tag{186}$$

Table 43. CA of Trained Elman RNNs

	Sufficient $N=30$		Not Sufficient $N=22$	
	$\overline{CA}$	$S_{\overline{CA}}$	$\overline{CA}$	$S_{\overline{CA}}$
Training Set	98.42%	0.11%	72.25%	2.27%
Validation Set	97.73%	0.30%	67.75%	2.57%
Test Set	98.82%	0.12%	60.12%	3.72%

Unfolded Layer	$\overline{\Gamma}_{noise}$	$S_{\overline{\Gamma}_{noise}}$	$\overline{\Gamma}_x$	$S_{\overline{\Gamma}_x}$	$\overline{\Gamma}_{y_{max}}$	$S_{\overline{\Gamma}_{y_{max}}}$	$\overline{\Gamma}_{y_{min}}$	$S_{\overline{\Gamma}_{y_{min}}}$
0	43.65	5.52	602.15	25.11	496.10	14.43	464.56	13.79
1	16.53	2.99	177.33	29.17	201.34	18.15	183.57	15.87
2	10.01	1.76	92.36	18.15	127.66	11.32	122.70	11.70
3	3.96	1.37	36.98	11.15	45.48	11.01	41.91	9.78
4	1.90	0.95	13.20	6.02	18.32	7.78	16.65	6.94
5	1.02	0.54	6.63	3.26	6.16	3.56	5.52	3.21
6	0.38	0.23	3.31	1.79	1.63	0.88	1.44	0.77
7	0.18	0.15	1.72	1.25	0.26	0.13	0.19	0.09
8	0.11	0.10	1.13	0.86	0.08	0.05	0.08	0.06

$$\overline{\Gamma}^1_{y_{min}} = \frac{\sum_{n=1}^{N} \min\{\Gamma^1_{y_1}(n), \Gamma^1_{y_2}(n)\}}{N} \tag{187}$$

where $\overline{\Gamma}^1_{y_{max}}$ is the partial derivative with respect to the maximum context node on unfolded layer 1 over $N = 30$ sufficiently trained Elman RNNs, $\Gamma^1_{y_{j^1}}(n)$ is the realized partial derivative with respect to context node y_{j^1} for $j^1 = 1,2$ on unfolded layer 1 for the nth sufficiently trained Elman RNN, and $\overline{\Gamma}^1_{y_{min}}$ is the partial derivative with respect to the minimum context node on unfolded layer 1 over $N = 30$ sufficiently trained Elman RNNs. It is necessary to calculate the averages for the context nodes in this fashion due to the *flip-flop* reversing nature of the trained weights associated with context nodes. When several ANNs are trained and then compared to each other, the roles of the hidden/context nodes often reverse. As an example, consider an Elman RNN with two hidden/context nodes and one output node. For the weights resulting from the first training session, it is possible that hidden/context node 1 fires high when the desired output is 1 and fires low when the desired output is -1. Hidden/context node 2 fires low

when the desired output is 1 and fires high when the desired output is -1. The next time the Elman RNN is trained, it is possible that the weights do a *flip-flop*. In other words, the weights that were associated with hidden/context node 1 are now associated with hidden/context node 2 and vice versa. So now hidden/context node 1 fires low when the desired output is 1 and fires high when the desired output is -1. And now hidden/context node 2 fires high when the desired output is 1 and fires low when the desired output is -1. Because of this *flip-flop* of the weights associated with the hidden/context nodes, care must be taken when computing the average of the partial derivative-based saliency measure of the hidden/context nodes. In the case of two hidden/context nodes, the average is taken for the maximum and the minimum. Otherwise, the averages of the partial derivative base saliency measure of the hidden/context nodes, in the limit, would be the same.

Figure 72 provides plots of the average partial derivatives in addition to 95% confidence intervals on the averages. These plots show a decline in the average partial derivatives for each additional unfolded layer. This, in essence, means that more current or recent values of the input x and context nodes y_j strongly effect the current output z.

One-sided t-tests with $\alpha = 0.05$ were run in order to determine the layer ℓ at which the input and context nodes on layer ℓ provide no more information than noise on layer ℓ. One-sided t-tests were also run in order to compare the input and context nodes to each other.
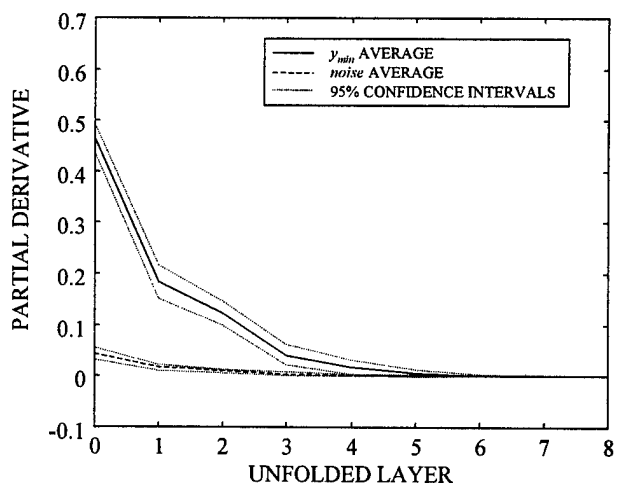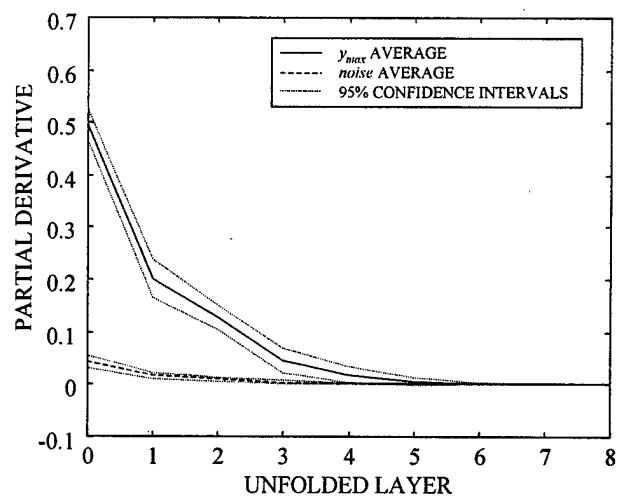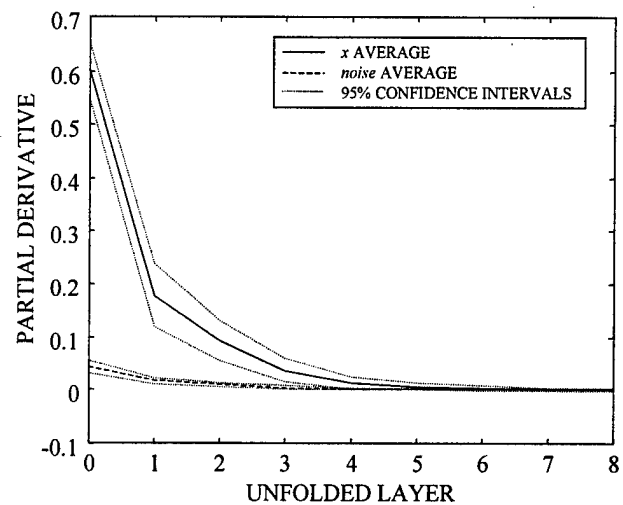
Figure 72. Partial Derivatives Over Time

In the case of comparing the partial derivative of the input x to *noise* at unfolded layer 1, the null hypothesis H_0 states that there is no significant difference between the average partial derivative for x at unfolded layer 1 denoted as $\overline{\Gamma}_x^1$ and that of *noise* denoted as $\overline{\Gamma}_{noise}^1$. The alternate hypothesis H_a states that $\overline{\Gamma}_x^1$ is either lower or higher than $\overline{\Gamma}_{noise}^1$. The test statistic assuming independent pairwise samples and assuming the variance of the two samples is unknown and unequal is calculated as:

$$t_s = \frac{\overline{\Gamma}_x^1 - \overline{\Gamma}_{noise}^1}{\sqrt{\dfrac{S_{\Gamma_x^1}^2 + S_{\Gamma_{noise}^1}^2}{N}}} \tag{188}$$

where t_s is the $t-$test statistic, N is the number of trained Elman RNNs ($N = 30$), $S_{\Gamma_x^1}^2$ is the sample variance of Γ_x^1 for $n = 1, ..., N$, and $S_{\Gamma_{noise}^1}^2$ is the sample variance of Γ_{noise}^1 for $n = 1, ..., N$ [88]. H_0 is rejected if $|t_s| \geq t_{\alpha, N-1}$ where $t_{\alpha, N-1}$ is the critical $t-$value for a given level of significance α and degrees of freedom $N-1$. The level of significance was set to $\alpha = 0.05$ and there were $N - 1 = 29$ degrees of freedom. As such, $t_{\alpha, N-1} = t_{0.05, 29} = 1.6991$. In the case where H_0 is rejected and $t_s < -t_{\alpha, N-1}$, the test concludes that $\overline{\Gamma}_x^1$ is lower than $\overline{\Gamma}_{noise}^1$. In the case where H_0 is rejected and $t_s > t_{\alpha, N-1}$, the test concludes that $\overline{\Gamma}_x^1$ is higher than $\overline{\Gamma}_{noise}^1$. Table 45 lists the test statistics. Those comparisons that are significant are shaded. The input and context nodes provide more information than noise up to the fourth unfolded layer. We now know that the following inputs, on average, significantly effect $z(t)$: $x(t)$, $x(t-1)$, $x(t-2)$, $x(t-3)$, and $x(t-4)$. We now also know that the following context nodes, on average, significantly

270

Table 45. Calculated t – Statistics

	Unfolded Layer	x	$y_{\max}$	$y_{\min}$
noise	0	21.3695	28.7895	27.8576
	1	5.3916	9.8800	10.1719
	2	4.4414	10.0985	9.3629
	3	2.8893	3.6795	3.7781
	4	1.8255	2.0592	2.0716
	5	1.6698	1.4017	1.3581
	6	1.5955	1.3542	1.2942
	7	1.19	0.3631	0.0698
	8	1.1593	0.3324	0.3047
x	0	*	3.6002	4.7224
	1	*	-0.6872	-0.1848
	2	*	-1.6228	-1.3814
	3	*	-0.5330	-0.3267
	4	*	-0.5109	-0.3690
	5	*	0.0958	0.2383
	6	*	0.8258	0.9417
	7	*	1.1407	1.1931
	8	*	1.2078	1.2046
$y_{\max}$	0	*	*	1.5538
	1	*	*	0.7248
	2	*	*	0.2996
	3	*	*	0.2381
	4	*	*	0.1570
	5	*	*	0.1308
	6	*	*	0.1611
	7	*	*	0.3850
	8	*	*	-0.0309

effect $z(t)$: $y_j(t-1)$, $y_j(t-2)$, $y_j(t-3)$, $y_j(t-4)$, and $y_j(t-5)$ for $j=1,2$. The memory capacity of an Elman RNN with an architecture as that in Figure 70 and Figure 71 for this wave amplitude problem is four unfolded layers.

271

For unfolded layer 0, the input $x(t)$ significantly effects $z(t)$ more, on average, than the context nodes $y_j(t-1)$ for $j = 1, 2$.

9.5 Conclusions

This methodology for determining the *memory capacity* of an Elman RNN provides insight into the theoretical workings of RNNs. It is now possible to calculate how far back in time, on average, an Elman RNN *remembers* for a given data set, a given Elman RNN architecture, and a given noise distribution to the extent that it is appropriate to measure *memory* by the partial derivative-based saliency measure over time. The significant contribution of this chapter was the theoretical development of a technique to determine the memory capacity of an Elman RNN that was applied to a wave amplitude detection problem.

10 Conclusions and Recommendations

10.1 Introduction

This chapter summarizes the significant contributions resulting from this research. This chapter also provides recommendations for future research.

10.2 Significant Contributions

This research resulted in three referee-reviewed conference papers [46,47,48]and two submitted archival journal papers [49,50]. This section summarizes the significant contributions resulting from this research.

10.2.1 Development of the Signal-to-Noise Ratio (SNR) Saliency Measure in Feedforward Multilayer (MLP) Artificial Neural Networks (ANN) to Classify Pilot Workload and Air Traffic Controller Workload

The development of the SNR saliency measure in feedforward ANNs has resulted in one referee-reviewed conference paper for classifying pilot workload [46] and one submitted archival journal paper for classifying air traffic controller workload [50]. The referee-reviewed conference paper for classifying pilot workload was selected as second runner-up for best conference paper with a novel engineering application at the 1996 Artificial Neural Networks in Engineering (ANNIE) Conference, St. Louis, MO.

10.2.2 Empirical Evidence that the Signal-to-Noise Ratio (SNR) Saliency Measure Provides Rankings Consistent with that of Other Saliency Measures

This dissertation provided empirical evidence that the SNR saliency measure

provides rankings consistent with that of a derivative-based saliency measure [124,126] and a weight-based saliency measure [152].

10.2.3 Development of the Signal-to-Noise Ratio (SNR) Screening Method in Feedforward Multilayer Perceptron (MLP) Artificial Neural Networks (ANN) to Classify Pilot Workload

The development of the SNR screening method in a feedforward MLP ANN to classify has resulted in an archival journal paper to appear in *Neurocomputing* [49].

10.2.4 Development of the Signal-to-Noise Ratio (SNR) Screening Method in Elman Recurrent Neural Networks (RNN) to Estimate Pilot Workload

The development of the SNR screening method in an Elman RNN to estimate pilot workload has resulted in a referee-reviewed conference paper [47].

10.2.5 Development of a Partial Derivative-Based Spatial-Temporal Screening Method for Elman Recurrent Neural Networks (RNN)

One archival journal paper will be submitted summarizing the development of a partial derivative-based spatial-temporal screening method for Elman RNNs.

10.2.6 Development of a Methodology For Determining the Memory Capacity of an Elman Recurrent Neural Network (RNN)

The development of a methodology for determining the memory capacity of an Elman RNN resulted in referee-reviewed conference paper [48]. The conference paper

was selected as second runner-up for best conference paper with a theoretical development in technique at the 1998 Artificial Neural Networks in Engineering (ANNIE) Conference, St. Louis, MO. One archival journal paper will be submitted summarizing the development of a methodology for determining the memory capacity of an Elman RNN.

10.3 Recommendations for Future Research

There are many areas for future research and this section will list but just a few.

10.3.1 Distribution of the Signal-to-Noise Ratio (SNR) Saliency Measure

It is desirable to determine the distribution of the SNR saliency measure and then utilize this distribution when testing for saliency. White concludes that the weights of a feedforward ANN, under certain assumptions, are distributed normally [163]. Assuming the weights are normally distributed, then the numerator $\sum_{j=1}^{J}\left(w_{i,j}^{1}\right)^{2}$ and the denominator $\sum_{j=1}^{J}\left(w_{N,j}^{1}\right)^{2}$ of the ratio may both have χ^{2} distributions with J degrees of freedom [88].

If both the numerator and the denominator have χ^{2} distributions, then the following ratio

$$\frac{\sum_{j=1}^{J}\left(w_{i,j}^{1}\right)^{2}}{\sum_{j=1}^{J}\left(w_{N,j}^{1}\right)^{2}} \tag{189}$$

may have a F distribution with J numerator degrees of freedom and J denominator degrees of freedom [88].

275

10.3.2 Distribution of the Injected Noise Feature

Another future research area may be determining the appropriate distribution of the injected noise feature. This research used a uniform distribution for noise. However, there are other distributions of noise that may be used such as Gaussian noise. The form of normalization may effect the optimal distribution of noise to use. For example, if the features are normalized between 0 and 1, then this research used a uniform(0,1) distribution for the injected noise feature. If the features are normalized between -1 and 1, then this research used a uniform(-1,1) distribution for the injected noise feature. However, what distribution of noise should be used if the features are standardized instead of normalized? Would it be more optimal to use Gaussian noise in this case?

10.3.3 Use of Saliency Measures in Architecture Selection

Future research may investigate the use of the partial derivative-based saliency measure and the SNR saliency measure for use in architecture selection for determining the optimal number of hidden nodes in a feedforward ANN. Research in this direction using the SNR saliency measure was initiated by Rizzo but much more research is needed [113]. Issues to be addressed regarding the SNR saliency measure include figuring out where the important information for deriving hidden node saliency is located in the ANN. In other words, is information for deriving the saliency of a hidden node found in the first layer weights, the second layer weights, or both?

10.3.4 Other Types of Recurrent Neural Networks (RNN)

All of the RNN research done in this dissertation was accomplished using an

276

Elman RNN. There are two other popular RNNs in the literature: the Jordan RNN [69] and the Williams and Zipser RNN [166,167]. Whereas the Elman RNN feeds back the hidden layer onto the input layer, the Jordan RNN feeds back the output layer onto the input layer. The Williams and Zipser RNN is a combination of the Elman RNN and the Jordan RNN in that it feeds back both the hidden and output layers onto the input layer. Future research may utilize the partial derivative-based saliency measure over time for feature saliency in the Jordan RNN and the Williams and Zipser RNN. Additional research may utilize the partial derivative-based saliency measure over time for determining the memory capacity of a Jordan RNN and a Williams and Zipser RNN.

10.3.5 User Friendly Software Development

It is highly recommended that user friendly software similar to that of Reinhart be developed to perform the methodologies as developed in this research [111]. *Matlab* and its *Neural Network Toolbox* are recommended to perform the algorithms. The user interface needs to be improved. The *Matlab* code written for this dissertation should be modified to be more general and allow for many modifications that the user would input in a simple but efficient manner. Parameters that the user may modify could include the various transfer functions (i.e. sigmoid or hyperbolic tangent), number of hidden nodes, number of training epochs, type of neural network (i.e. feedforward MLP ANN or RNN), et cetera.

10.3.6 Address Individual Differences in Workload

Research is already being started to select a parsimonious set of salient

psychophysiological features for classifying air traffic controller workload using the SNR saliency measure and its associated screening method. This research being conducted by Laine and will attempt to determine if one ANN classifier is sufficient to classify air traffic controller or if each air traffic controller will require his own ANN classifier [75]. Approximately 10-12 test subjects will be utilized.

10.3.7 In-Flight Pilot Workload Data Collection

Future research must involve extensive data collection efforts in order to obtain in-flight pilot workload data. As a minimum, three test flights with data collection is necessary for each test subject pilot since it is highly desirable to have a training, test, and validation set. Due to the temporal nature of the RNNs, the training, test, and validation data sets should be collected on separate but near identical test flights. Various issues will arise with in-flight data collection which will require investigation to include noise, vibration, G-forces, and EEG artifacts. As a minimum, data should be collected for twelve test subject pilots. With an extensive in-flight data set, research for determining a parsimonious set of salient features may be conducted on pilot workload. In addition, individual differences in pilot workload may be investigated for determining if a robust parsimonious set of salient features exists for both feedforward ANNs and the three types of RNNs.

1.3.8 Investigate Other Electroencephalography (EEG) Preprocessors

The only preprocessors for EEG is the FFT and the elliptic filter. Other possibilities that show promise include feature space trajectory neural networks (FSTNN)

and wavelets.

A FSTNN allows for shift-invariance and distortion-invariance by representing a feature vector as a trajectory in the feature space [99]. Calculation of the closest feature space trajectory results in classification. A FSTNN can be used in speech recognition where words represent a sequence of phonemes [16]. The major advantage to a FSTNN is its success in classify overlapping data sets since its decision rule is based on distance to a class trajectory [16]. A FSTNN is similar to a nearest neighbor algorithm. In a FSTNN, a test point X is inputted and the vector inner product with each of the "links" in the trajectory is calculated to find the closest point P on the trajectory to X [16]. The closest link wins. Brandstrom used a FSTNN to compare sequences of images of satellites passing over an observatory on Maui to known sequences to identify the satellite and its orbit.

A dynamic time warp (DTW) in conjunction with the FSTNN may be an option for preprocessing the EEG data to encode temporal information. The DTW algorithm provides a way to incorporate sequence information without increasing the dimensionality of the network because the sequence information is encoded in the algorithm itself rather than additional features [16]. The DTW does not allow for a test point X to project to a link more previous in the sequence than the link selected by the previous test point [16]. Ideally, each subsequent test point will project to sequential links in the design trajectory [Bruegger]. A point may skip ahead two links instead of one but the distance calculation will be penalized [16]. The penalty is the product of the distances to non-preferred links and a "stretch factor" [16]. Ney used a DTW for recognizing connected speech.

Wavelets transform a signal into a sum of small, overlapping waves [145]. The wavelet transform is the newest way to analyze and synthesize a signal (the Fourier transform and the FFT are the others). In Fourier analysis, the signal becomes the sum of cosine waves. The major disadvantage to the Fourier transform is that its cosines go on forever [145]. With the FFT, segments of the signal are transformed into cosines separately. The FFT therefore allows for each segment to have a different amplitude. The major disadvantage to the FFT is the sudden discontinuity between segments or rather its "blocking effect" [145]. The smoothing over a 10-second window in the preprocessing of the EEG signal is an attempt to correct for this.

Instead of cosine building blocks, a wavelet transform has a wavelet building block. A wavelet is a small wave that starts and stops. Each wavelet in a transform comes from a "mother wavelet" $W(t)$ [145]. The mathematical framework for the wavelet transform was developed by Mallet. The framework is based on the notion of a multiresolution analysis consisting of approximating vector spaces V_j where $j \in Z$ and a scaling function ϕ [72]. The set of functions $\left\{ 2^{\frac{j}{2}} \phi(2^j x - n) \middle| n \in Z \right\}$ forms *an orthonormal basis for V_j [72].

Bibliography

1. Altenmuller, E.O. "Psychophysiology and EEG," *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, Third Edition, eds. E. Niedermeyer and F. Lopes da Silva. Baltimore, MD: Williams & Wilkins, 1993, pp. 597-613.

2. Apostol, T.M. *Mathematical Analysis*, Second Edition, Reading, MA: Addison-Wesley Publishing, 1974.

3. Auten, J. "G-LOC. Is the Cluebag Half Full or Half Empty?" *Torch*, Sept 1995, pp. 8-11.

4. Ayres, F. and E. Mendelson. *Schaum's Outline of Theory and Problems of Differential and Integral Calculus*, Third Edition, St. Louis, MO: McGraw-Hill, 1990.

5. Barnsley, M.F. *Fractals Everywhere*, Second Edition, Boston, MA: Academic Press Professional, 1993.

6. Bauer, K.W. Personal conversations, 1995-1998. Also see "Preface" of [147].

7. Bauer, L.O., Goldstein, R., and Stern, J.A. "Effects of Information - Processing Demands on Psychophysiological Response Patterns," *Human Factors*, Vol 29, 1987, pp. 213-234.

8. Bauer, L.O., Strock, B.D., Goldstein, R., Stern, J.A., and Walrath, L.C. "Auditory Discrimination and the Eye Blink," *Psychophysiology*, Vol 22, 1984, pp. 636-641.

9. Beatty, J. "Phasic Not Tonic Pupillary Responses Vary with Auditory Vigilance Performance," *Psychophysiology*, Vol 19, 1982, pp. 167-172.

10. Beideman, L.R. and Stern, J.A. "Aspects of the Eye Blink During Simulated Driving as a Function of Alcohol," *Human Factors*, Vol 19, 1977, pp. 73-77.

11. Belue, L.M. *Multilayer Perceptrons for Classification*, M.S. Thesis, Air Force Institute Of Technology, OH, Mar 1992.

12. Belue, L.M. and K.W. Bauer, "Determining Input Features For Multilayer Perceptrons," *Neurocomputing*, Vol 7, 1995, pp. 111-121.

13. Berger, H. "Uber das Elektrenkephalogramm des Menschen," *Archives of Psychology*, Vol 28, 1929, pp. 527-570.

14. Brandstrom, G.W. *Space Object Identification using Spatio-Temporal Pattern Recognition*, M.S. Thesis, Air Force Institute of Technology, 1995.

15. Brookings, J.B., Wilson, G.F., and Swain, C.R. "Psychophysiological Responses to Changes in Workload During Simulated Air Traffic Control," *Biological Psychology*, Vol 42, 1996, pp. 361-378.

16. Bruegger, N.W. *Feature Space Trajectory Neural Networks*, M.S. Thesis, Air Force Institute of Technology, Mar 1997.

17. Caldwell, J.A., Kelly, C.F., Roberts, K.A., Jones, H.D., Lewis, J.A., Woodrum, L., Dillard, R.M., and Johnson, P.P. *A Comparison of EEG and Evoked Response Data Collected in a UH-1 Helicopter to Data Collected in a Standard Laboratory Environment*, USAARL Report No 97-30, Aug 1997.

18. Caldwell, J.A., Lewis, J.A., Darling, S.R., Dillard, R.M., and Johnson, P.P. *Collection of Real-time, Multichannel EEG Data from Helicopter Pilots in Flight: A Feasibility Study*, USAARL Report No 94-26, May 1994.

19. Caldwell, J.A., Roberts, K.A., Kelly, C.F., Jones, H.D., Lewis, J.A., Woodrum, L, Dillard, R.M., and Johnson, P.P. *Effects of Pilot Workload on EEG Activity Recorded During the Performance of In-Flight Maneuvers in a UH-1 Helicopter*, USAARL Report No 97-31, Aug 1997.

20. Caldwell, J.A., Wilson, G.F., Centiguc, M., Gaillard, A.W.K., Gundel, A., Lagarde, D., Makeig, S., Myhre, G., and Wright, N.A. *Psychophysiological Assessment Methods*, Advisory Group for Aerospace Research and Development (AGARD) Advisory Report 324. Paris, France: AGARD, 1994.

21. Casali, J.G. and W.W. Wierwille. "A Comparison of Rating Scale, Secondary-Task, and Physiological, and Primary-Task Workload Estimation Techniques in a Simulated Flight Task Emphasizing Communications Load," *Human Factors*, Vol 25, 1983, pp. 623-641.

22. Corkindale, K.G., Cumming, F.G., and Hammerton-Fraser, A.M. "Physiological Assessment of Pilot Stress During Landing," *Measurement of Aircrew Performance Conference*, Brooks Air Force Base (AFB), TX, AGARD Conference Proceeding No 56, 1983.

23. Cover, T.M. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications to Pattern Recognition," *IEEE Transactions Electronic Computers*, Vol EC-14, Jun 1965, pp. 326-334.

24. Demuth, H. and M. Beale. *MATLAB Neural Network Toolbox User's Guide*, Natick, MA: MathWorks, 1998.

25. Devijver, P.A. and J. Kittler. *Pattern Recognition: A Statistical Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1982.

26. Dillon, W.R. and M. Goldstein. *Multivariate Analysis: Methods and Applications*, New York, NY: John Wiley & Sons, 1984.

27. Donchin, E., Ritter, W., and McCallum, W.C. "Cognitive Psychophysiology: The Endogenous Components of the ERP," *Event-Related Potentials in Man*, eds. E. Callaway, P. Tueting, and S.H. Coslow, New York, NY: Academic Press, 1978, pp. 349-411.

28. Duda, R.O. and P.E. Hart. *Pattern Classification and Scene Analysis*, New York, NY: John Wiley and Sons, 1973.

29. Duffy, E. "Activation," *Handbook of Psychophysiology*, eds. N.S. Greenfield and R.A. Steinbach, New York, NY: Holt, Rinehart, and Winston, 1972, pp. 577-622.

30. Eason, R.G., Beardshall, A., and Jaffe, S. "Performance and Physiological Indicates of Activation in Vigilance Situation," *Perceptual and Motor Skills*, Vol 20, 1965, pp. 3-13.

31. Elman, J.L. "Finding Structure in Time," *Cognitive Science*, Vol 14, 1990, pp. 179-211.

32. Elul, R. "The Genesis of the EEG." *International Review of Neurobiololgy*, Vol 15, 1972, pp. 227-272.

33. Farmer, J.D. and J.J Sidorowich. *Exploiting Chaos to Predict the Future and Reduce Noise*, Version 1.1, Report No LA-UR-88-901, Los Alamos, NM: Los Alamos National Laboratory, Feb 1988.

34. Federal Aviation Administration (FAA). *National Airspace Plan: Facilities, Equipment, and Associated Development*. Washington, DC: FAA, 1985.

35. Fisher, R.A. "The Use of Multiple Measurements in Taxonomic Problems," *Annuls of Eugenis*, Vol 7, Part 2, 1936, pp. 179-188.

36. Fix, E.L. *Neural Network Based Human Performance Modeling*. Armstrong Laboratory Final Report, AAMRL-TR-90-042, 1990.

37. Foley, D.H. "Considerations of Sample and Feature Size," *IEEE Transactions on Information Theory*, Vol IT-18, No 5, Sep 1972, pp. 618-626.

38. Froehling, H. et al. "On Determining the Dimension of Fractal Flows," *Physica D*, Vol 3, 1981, pp. 605-617.

39. Fukunaga, K. *Introduction to Statistical Pattern Recognition*, Second Edition, New York, NY: Academic Press, 1990.

40. Gainey, J.C., Jr. *Predicting Nonlinear Time Series.* M.S. Thesis, Air Force Institute of Technology, 1993.

41. Gevins, A.S., Zeitlin, G.M., Doyle, J.C., Yingling, C.D., Schaffer, R.E., Callaway, E., and Yeager, C.L. "Electroencephalogram Correlates of Higher Cortical Functions," *Science*, Vol 203, 1979, pp. 665-668.

42. Gevins, A.S., Zeitlin, G.M., Yingling, C.D., Doyle, J.C., Dedon, M.F., Schaffer, R.E., Roumasset, J.T., and Yeager, C.L. "EEG Patterns During 'Cognitive' Tasks. I. Methodology and Analysis of Complex Behaviors," *Electroencephalography and Clinical Neurophysiology*, Vol 47, 1979, pp. 693-703.

43. Giannitrapani, D. *The Electrophysiology of Intellectual* Functions, New York, NY: Karger, 1985.

44. Goldstein, R., Walrath, L.C., Stern, J.A., and Strock, B.D. "Blink Activity in a Discrimination Task as a Function of Stimulus Modality and Schedule Presentation," *Psychophysiology*, Vol 22, 1985, pp. 629-635.

45. Grassberger, P. and I. Procaccia. "Measuring the Strangeness of Strange Attractors," *Physica D*, Vol 9, 1983, pp. 189-208.

46. Greene, K.A., Bauer, K.W., Kabrisky, M., Rogers, S.K., Russell, C.A., and Wilson, G.F. "A Preliminary Investigation of Selection of EEG and Psychophysiological Features for Classifying Pilot Workload," *Intelligent Engineering Systems through Artificial Neural Networks,* Proceedings of Artificial Neural Networks in Engineering (ANNIE) International Conference, St. Louis, MO, 10-13 Nov 1996, Vol 6, pp. 691-697.

47. Greene, K.A., Bauer, K.W., Kabrisky, M., Rogers, S.K., and Wilson, G.F. "Estimating Pilot Workload Using Elman Recurrent Neural Networks: A Preliminary Investigation," *Intelligent Engineering Systems through Artificial Neural Networks,* Proceedings of Artificial Neural Networks in Engineering (ANNIE) International Conference, St. Louis, MO, 9-12 Nov 1997, Vol 7, pp. 703-708.

48. Greene, K.A., Bauer, K.W., Kabrisky, M., Rogers, S.K., and Wilson, G.F. "Determining the Memory Capacity of an Elman Recurrent Neural Network," *Intelligent Engineering Systems through Artificial Neural Networks,* Proceedings of Artificial Neural Networks in Engineering (ANNIE) International Conference, St. Louis, MO, 1-4 Nov 1998, Vol 8, to appear.

49. Greene, K.A., Bauer, K.W., and Sumrell, D.B. "Feature Screening Using Signal-to-Noise Ratios," accepted by *Neurocomputing*, Aug 98.

50. Greene, K.A., Bauer, K.W., Wilson, G.F., Russell, C.A., Rogers, S.K., and Kabrisky, M. "Selection of Psychophysiological Features for Classifying Air Traffic Controller

Workload in Neural Networks," submitted to *International Journal of Smart Engineering System Design.*

51. Gundel, A., Drescher, J., Maab, H., Samel A., and Vejvoda, M. "Sleepiness of Civil Airline Pilots During Two Consecutive Night Flights of Extended Duration," *Biological Psychology Biological Psychology,* Vol 20, 1995, pp. 131-141.

52. Hancock, P.A. "Task Categorization and the Limits of Human Performance in Extreme Heat," *Aviation, Space and Environmental Medicine,* Vol 53, 1982, pp. 778-784.

53. Harris, R.L., Tole, J.R., Stephens, A.T., and Ephrath, A.R. "Visual Scanning Behavior and Pilot Workload," *Aviation, Space, and Environmental Medicine,* Vol 53, 1982, pp. 1067-1072.

54. Harmony, T., Fernandez, Silva J., Bernal, J., Diaz-Comas, L., Reyes, A., Marosi, E., Rodriguez, M., and Rodriguez, M. "EEG Delta Activity: An Indicator of Attention to Internal Processing during Performance of Mental Tasks," *International Journal of Psychophysiology,* Vol 24, 1996, pp. 161-171.

55. Hart, S.G., Hauser, J.R., and Lester, P.T. "Inflight Evaluation of Four Measures of Pilot Workload," *Proceedings of the Human Factors Society's 28th Meeting,* 1984, pp. 945-949.

56. Hart, S.G. and L.E. Staveland. "Development of the NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *Human Mental Workload,* Ed. P.A. Hancock and N. Meshkati, Amsterdam: North Holland, 1988, pp. 139-183.

57. Hatch, J.P., Klatt, K., Porges, S.W., Schroeder-Jasheway, L., and Supik, J.D. "The Relation Between Rhythmic Cardiovascular Variability and Reactivity to Orthostatic, Cognitive, and Cold Pressor Stress," *Psychophysiology,* Vol 23, 1986, pp. 48-56.

58. Hecht-Nielsen, R. *Neurocomputing.* New York: Addison-Wesley, 1991.

59. Highleyman, W.H. "Linear Decision Functions, with Application to Pattern Recognition," *Proceedings of the Institute of Radio Engineers,* Vol 50, No 6, Part 1, June 1962, pp. 1501-1514.

60. Hillyard, S.A. and J.C. Hansen. "Attention: Electrophysiological Approaches," *Psychophysiology: Systems, Processes, and Applications,* eds. M.G.H. Coles, E. Donchin, and S.W. Porges, New York NY: Guilford Press, 1986, pp. 227-243.

61. Hohnsbein, J., Falkenstein, M., and Horrman, J. "Effects of Attention and Pressure on P300 Components and Implications for Mental Workload Research," *Biological Psychology,* Vol 40, 1995, pp. 73-81.

285

62. Hornik, K., Stinchcombe, and White, H. "Multi-layer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol 2, May 1989, pp. 359-366.

63. Horst, Richard L. "An Overview of Current Approaches and Future Challenges in Physiological Monitoring," *Proceedings of the 1987 NASA Mental-State Estimation Workshop*, Williamsburg, VA. NASA Conference Publication 2504, 3-4 Jun 1987, pp. 25-42.

64. Ingvar, D.H. and J. Risberg. "Increase in Regional Cerebral Blood Flow During Mental Effort in Normals and in Patients with Focal Brain Disorders," *Experimental Brain Research*, Vol 3, 1967, pp. 195-211.

65. Isaksson, A. and A. Wennberg. "Spectral Properties of Nonstationary EEG Signals Evaluated by Means of Kalman Filtering: Application Examples From a Vigilance Test," *Quantitative Analytical Studies in Epilepsy*, eds. P. Kellaway and I. Peterson, New York, NY: Raven, 1976, pp. 389-402.

66. Jacquy, J., Noel, P., Segers, A., Huvelle, R., and Noel, G. "Regional Cerebral Blood Flow in Children: A Rheoencephalographic Study of Modifications Induced By Reading," *Electroencephalography and Clinical Neurophysiology*, Vol 42, 1977, pp. 691-697.

67. Jex, H.R., and R.W. Allen. "Research on a New Human Dynamic Response Test Battery: Part II. Psychophysiological Correlates," *Proceedings of the Sixth Annual Conference on Manual Control*, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1970, pp. 743-777.

68. John, E.R. and P. Easton. "Quantitative Electrophysiological Studies of Mental Tasks," *Biological Psychology*, Vol 40, 1995, pp. 101-113.

69. Jordan, M.I. *Serial Order: A Parallel Distributed Processing Approach*. ICS Report 8604. La Jolla, CA: Institute for Cognitive Science, May 1986.

70. Kerchner, R.M. and G.F. Corcoran. *Alternating Current Circuits*, Second Edition, New York, NY: John Wiley & Sons, 1943.

71. Kittler, J. and P.C. Young. "A New Approach to Feature Selection Based on the Karhunen-Loève Expansion," *Pattern Recognition*, Vol 5, 1973, pp. 335-352.

72. Kocur, C.M., Rogers, S.K., Myers, L.R., Burns, T., Kabrisky, M., Hoffmeister, J.W., Bauer. K.W., and Steppe, J.M. "Using Neural Networks to Select Wavelet Features for Breast Cancer Diagnosis," *IEEE Engineering in Medicine and Biology*, Vol 1080, May/June 1996, pp. 95-102.

73. Kramer, A.F., Trejo, L.J., and Humphrey, D. "Assessment of Mental Workload with Task-Irrelevant Auditory Probes," *Biological Psychology*, Vol 40, 1995, pp. 83-100.

74. Lacey, J.I. "Somatic Response Patterning and Stress: Some Revisions of Activation Theory," *Psychological Stress*, eds. M.H. Appley and R. Trumbell, New York, NY: Appleton-Century-Crofts, 1967.

75. Laine, T. M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, Mar 1999, *to appear*.

76. Lapedes, A. and R. Farber. *How Neural Networks Work*. Los Alamos National Laboratory Final Report, LA-UR-88-418, Jul 1987.

77. Lapedes, A. and R. Farber. "How Neural Networks Work," *Proceedings of 1987 IEEE Conference on Neural Information Processing Systems*, Denver, CO.

78. Lapedes, A. and R. Farber. *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling*, Los Alamos National Laboratory Final Report, LA-UR-87-2662, Jan 1988.

79. Lindsey, R.L. *Function Prediction Using Recurrent Neural Networks*, M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, Dec 1991.

80. Lindsley, D.B. "Psychological Phenomena and the Electroencephalogram," *Electroencephalography and Clinical Neurophysiology*, Vol 4, 1952, pp. 443-456.

81. Lippman, R.P. "An Introduction to Computing with Neural Nets," *IEEE Acoustics, Speech, and Signal Processing*, Vol 4, No 2, Apr 1987, pp. 4-22.

82. Mallat, S.G. "Multiresolution Approximation and Wavelets," *Transactions of the American Mathematical Society*, Sept 1989, pp. 69-88.

83. Malmo, R.B. "Activation: A Neurophysiological Dimension," *Psychological Review*, Vol 66, 1959, pp. 367-386.

84. Mantel, N. "Why Stepdown Procedures in Variable Selection," *Technometrics*, Vol 12, No 3, Aug 1970, pp. 621-625.

85. MathWorks. *MATLAB Signal Processing Toolbox User's Guide*. Natick, MA: MathWorks, 1996.

86. McCarthy, G. and E. Donchin. "A Metric for Thought: A Comparison of P300 Latency and Reaction Times," *Science*, Vol 211, 1981, pp. 77-79.

87. McCulloch, W.S. and W. Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol 5, 1943, pp. 115-133.

88. Mendenhall, W., Wackerly, D.D., and Schaeffer, R.L. *Mathematical Statistics with Applications*. Boston, MA: PWS-Kent, 1990.

89. Milam, D.W. Personal conversations, 1995-1998.

90. Milnor, J. "On the Concept of an Attractor," *Comm. Math. Phys.*, Vol 92, 1985, pp. 177-195.

91. Minsky, M.L. and Papert, S. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, Mass.: MIT Press, 1969.

92. Minsky, M.L. and Papert, S. *Perceptrons: An Introduction to Computational Geometry (Expanded Version)*. Cambridge, Mass.: MIT Press, 1988.

93. Montgomery, L.D., Montgomery R.W., Gerth, W.A., and Guisado, R. "Rheoencephalographic and Electroencephalographic Analysis of Cognitive Workload," *Proceedings of the Third Symposium on Computer-Based Medical Systems*, ed. N.J. Piscataway, IEEE Computer Society, 1990, pp. 220-227.

94. Montgomery, L.D., Montgomery, R.W., and Guisado, R. "Rheoencephalographic and Electroencephalographic Measures of Cognitive Workload: Analytical Procedures," *Biological Psychology*, Vol 40, 1995, pp. 143-159.

95. Montgomery, R.W., Montgomery, L.D., and Guisado, R. "Cerebral Event Related Vascular Responses to Mental Work," *Aviation, Space, and Environmental Medicine*, Vol 62, 1991, pp. 457.

96. Morrison, R. and R.H. Wright. "ATC Control and Communications Problems: An Overview of Recent ASRS Data," *Proceedings of the Fifth International Symposium on Aviation Psychology*, Vol 4, 1989, pp. 29-45.

97. Mulder, G. "Sinus Arrhythmia and Mental Workload," *Mental Workload: It's Theory and Measurement*, ed. N. Moray, New York, NY: Plenum Press, 1979, pp. 327-344.

98. Munson, R.C., Horst, R.L., and Mahaffey, D.L. "Primary Task Event-Related Potentials Related to Different Aspects of Information Processing," *Proceedings of the 1987 NASA Mental-State Estimation Workshop*, Williamsburg, VA, 3-4 Jun 1987. NASA Conference Publication No 2504, pp. 163-178.

99. Neiberg, L., and Casasent, D.P. "Feature Space Trajectory Neural Net Classifier," *SPIE*, Vol 2492, 1995, pp. 361-372.

100. Neidermeyer, E. and F. Lopes da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, Third Edition, Baltimore, MD: Williams & Wilkins, 1993.

101. Netrella, M.G. *Experimental Statistics.* Washington DC: United States Department of Commerce, 1966.

102. Nguyen, D, and B. Widrow. "Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights," *International Joint Conference on Neural Networks*, Vol 3, 1990, pp. 21-26.

103. Nilsson, N.J. *Learning Machines: Foundations of Trainable Pattern Classifying Systems.* New York, NY: McGraw-Hill, 1965.

104. Okada, Y.C., Kaufman, L., and Williamson, S.J. "The Hippocampal Formation as a Source of Slow Endogenous Potentials," *Electroencephalography and Clinical Neurophysiology*, Vol 55, 1983, pp. 417-426.

105. Olds, E.G. "Distributions of Sums of Squares of Rank Differences for Small Numbers of Individuals," *Annuals of Mathematical Statistics*, Vol 9, 1938, pp. 133-148.

106. Persson, J. "Comments on Estimations and Tests of EEG Amplitude Distributions," *Electroencephalography and Clinical Neurophysiology*, Vol 46, 1979, pp. 309-313.

107. Piraux, A., Jacquy, J., Lhoas, J.P., Wilmotte, J., and Noel, G. "Regional Cerebral Blood Flow Variations in Mental Alertness," *Neuropsychobiology*, Vol 1, 1975, pp. 335-343.

108. Porgues, S.W. "Respiratory Sinus Arrhythmia: An Index of Vagal Tone," *Psychophysiology of Cardiovascular Control*, eds. J.F. Orlebeke, G. Mulder, and L.J. VanDoornen, New York, NY: Plenum Press, 1985, pp. 437-450.

109. Porgues, S.W. "Vagal Tone as an Index of Mental State," *Proceedings of the 1987 NASA Mental-State Estimation Workshop*, Williamsburg, VA, NASA Conference Publication 2504, 3-4 Jun 1987, pp. 57-64.

110. Reed, R. "Pruning Algorithms - A Survey," *IEEE Transactions on Neural Networks*, Vol 4, No 5, 1993, pp. 740-747.

111. Reinhart, G.L. *A Fortran Based Learning System using Multilayer Backpropagation Neural Network Techniques.* M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, Mar 1994.

112. Ritter, W., Simson, R., and Vaughan, H.G. "Association Cortex Potentials and Reaction Time in Auditory Discriminations," *Electroencephalography and Clinical Neurophysiology*, Vol 33, 1972, pp. 547-557.

113. Rizzo, C. *Parallel Implementation of an Artificial Neural Network Integrated Feature and Architecture Selection Algorithm.* M.S. Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, Mar 1998.

114. Robinson, E.R.N. "Biotechnology Predictors of Physical Security Personnel," *A Review of the Stress Literature Related to Performance*, Report NPRDC TN-83-9, San Diego, CA: Navy Personnel Research and Development Center, 1983.

115. Rogers, S.K. *Advanced Biological and Artificial Neural Networks*, SPIE, Apr 1996.

116. Rogers, S.K. Personal conversations, 1995-1998.

117. Roscoe, A.H. "In-Flight Assessment of Workload Using Pilot Ratings and Heart Rate," *The Practical Assessment of Pilot Workload*, ed. A.H. Roscoe, AGARDograph No. 282. Neuilly sur Seine, France: AGARD, 1987, pp. 1-14.

118. Roscoe, A.H. "Heart-Rate as an In-Flight Measure of Pilot Workload," *Proceedings of the Workshop on Flight Testing to Identify Pilot Workload and Pilot Dynamics*, Edwards AFB, CA, eds. M.L. Frazier and R.B. Crombie, AFTEC-TR-82-5, 1992, pp. 338-349.

119. Rosenblatt, F. *On the Convergence of Reinforcement Procedures in Simple Perceptrons.* Report VG-1196-G-4. Buffalo, NY: Cornell Aeronautical Laboratory, 15 Feb 1960.

120. Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanics.* Washington D.C.: Spartan Books, 1962.

121. Rosenblatt, F. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, Vol 65, No 6, 1958, pp. 386-408.

122. Rosenblatt, F. *The Perceptron: A Theory of Statistical Separability in Cognitive Systems.* Report VG-1196-G-1. Buffalo, NY: Cornell Aeronautical Laboratory, Jan 1958.

123. Rosenblatt, F. *Two Theorems of Statistical Separability in the Perceptron.* Report VG-1196-G-2. Buffalo, NY: Cornell Aeronautical Laboratory, 1 Sept 1958.

124. Ruck, D.W. *Characterization of Multilayer Perceptrons and their Application to Multisensor Automatic Target Detection*, Ph.D. Dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, Dec 1990.

125. Ruck, D.W. and S.K. Rogers. *Feature Selection for Pattern Recognition Using Multilayer Perceptrons*, Air Force Institute of Technology, Jun 1995.

126. Ruck, D.W., Rogers, S.K., and M. Kabrisky. "Feature Selection Using a Multilayer Perceptron," *Journal of Neural Network Computing*, Fall 1990, pp. 40-48.

127. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., and Suter, B.W. "The Multi-layer Perceptron as an Approximator to a Bayes Optimal Discriminant Function," *IEEE Transactions on Neural Networks*, Vol 1, No 4, December 1990, pp. 296-298.

128. Rumelhart, D.E., McClelland, J.L, and the PDP Research Group. *Parallel Distributing Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Cambridge, Mass.: MIT Press, 1986.

129. Russell, C.A., Wilson, G.F., and Monett, C.T. "Mental Workload Classification Using a Backpropagation Neural Network," *Intelligent Engineering Systems through Artificial Neural Networks,* Proceedings of Artificial Neural Networks in Engineering (ANNIE) International Conference, St. Louis, MO, 10-13 Nov 1996, eds. C.H. Dagli, M. Akay, C.L.P. Chen, B.R. Fernandez, and J. Ghosh, Vol 6, pp. 685-690.

130. Sayers, B. "Physiological Consequences of Informational Load and Overload," *Research in Physiology*, eds. P.H. Venables and M.J. Christie, New York, NY: Wiley, 1975, pp. 95-124.

131. Setiono, R. "A Penalty Function Approach for Pruning Feedforward Neural Networks," *Neural Computation*, Vol 9, No1, 1997, pp. 185-204.

132. Setiono, R. and H. Liu. "Neural Network Feature Selector," *IEEE Transactions on Neural Networks*, Vol 8, No 3, May 1997, pp. 654-662.

133. Shaudys, F.E. and T.K. Leen, "Feature Selection for Improved Classification," *International Joint Conference on Neural Networks*, Vol. 4, Baltimore, MD, 7-11 Jun 1992, pp. 697-702.

134. Skrandies, W. "Visual Information Processing: Topography of Brain Electrical Activity," *Biological Psychology*, Vol 40, 1995, pp. 1-15.

135. Speyer, J.J., Fort, A., Fouillot, J., and Blomberg, R.D. "Dynamic Methods for Assessing Workload for Minimum Crew Certification," *Workload in Transport Operations*, eds. A.H. Roscoe and H.C. Muir. DFVLR No 1B 316-88-06, Cologne, France: DFVLR, 1988, pp. 196-220.

136. Steppe, J.M. *Feature and Model Selection in Feedforward Neural Networks*, Ph.D. Dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 1994.

137. Steppe, J.M. and K.W. Bauer. "Feature Saliency Measures," *Proceedings of Ohio Aerospace Institute (OAI) Neural Network Symposium and Workshop*, Ohio University, OH, 21-22 August 1995, pp. 295-325.

138.   Steppe, J.M. and K.W. Bauer. "Feature Saliency Measures," *Computers & Mathematics with Applications*, Vol 33, No 8, 1997, pp. 109-126.

139.   Steppe, J.M. and K.W. Bauer. "Improved Feature Screening in Feedforward Neural Networks," *Neurocomputing*, Vol 13, 1996, pp. 47-58.

140.   Steppe, J.M., Bauer, K.W., and Rogers, S.K. "Integrated Feature and Architecture Selection," *IEEE Transactions on Neural Networks*, Vol 7, No 4, July 1996, pp. 1007-1014.

141.   Sterman, M.B., Kaiser, D.A., Mann, C.A., Suyenobu, B.Y., Beyma, D.C., and Francis, J.R. "Application of Quantitative EEG Analysis to Workload Assessment in an Advanced Aircraft Simulator," *Proceedings of the 1993 Human Factors and Ergonomics Society 37th Annual Meeting*, pp. 118-121.

142.   Sterman, M.B., and Mann, C.A. "Concepts and Applications of EEG Analysis in Aviation Performance Evaluation," *Biological Psychology*, Vol 40, 1995, pp. 115-130.

143.   Sterman, M.B., Mann, C.A., and Kaiser, D.A. "Qualitative EEG Patterns of Differential In-Flight Workload," *Space Operations Applications and Research Proceedings*, Johnson Space Center, Houston, TX, 1992, pp. 466-473.

144.   Stern, J.A. and D.N. Dunham. "The Ocular System," *Principles of Psychophysiology: Physical, Social, and Inferential Elements*, Cacioppo, J.T. and L.G. Tassinary (eds). Cambridge, MA: Cambridge University Press, 1990, pp. 513-553.

145.   Strang, G. and T. Nguyen. *Wavelets and Filter Banks*. Wellesley, MA: Wellesley-Cambridge Press, 1996.

146.   Stright, J.R. *A Neural Network Implementation of Chaotic Time Series Prediction*, Ph.D. Dissertation, Air Force Institute of Technology, OH, 1988.

147.   Sumrell, D.B. *An Investigation of Preliminary Feature Selection Using Signal-to-Noise Ratios*, M.S. Thesis, Air Force Institute of Technology, OH, 1996.

148.   Sutton, S., Braren, M., Zubin, J., and John, E.R. "Evoked Potential Correlates of Stimulus Uncertainty," *Science*, Vol 150, 1966, pp. 1187-1188.

149.   Swain, P.H. "Pattern Recognition Techniques in Remote Sensing Applications," *Handbook of Statistics: Classification, Pattern Recognition, and Reduction of Dimensionality*, Vol 2, eds. P.R. Krishnaiah and L.N. Kanal, New York, NY: North-Holland, 1982.

150.   Takens, F. "Detecting Strange Attractors in Turbulence" *Lecture Notes in Mathematics, Proceedings of Dynamical Systems and Turbulence Symposium*, University of Warwick, Scotland, Vol 898, 1981, pp. 366-381.

151.   Takens, F. "On the Numerical Determination of the Dimension of an Attractor," *Lecture Notes in Mathematics, Proceedings of Dynamical Systems and Bifurication Workshop*, Groningen, The Netherlands, Vol 1125, Apr 1984, pp. 99-106.

152.   Tarr, G.L. *Multilayered Feedforward Neural Networks for Image Segmentation*, Ph.D. Dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, 1991.

153.   Thurn, S.B. et al. *The MONK'S Problems - A Performance Comparison of Different Learning Algorithms*, Technical Report CMU-CS-91-197, Carnegie-Mellon University, PA, 1991.

154.   Trejo, L.J., Kramer, A.F., and Arnold, J.A. "Event-Related Potentials as Indices of Display-Monitoring Performance," *Biological Psychology*, Vol 40, 1995, pp. 33-71.

155.   Ullsperger, P. and K. Grune. "Processing of Multi-Dimensional Stimuli: P300 Component of the Event-Related Brain Potential During Mental Comparison of Compound Digits," *Biological Psychology*, Vol 40, 1995, pp. 17-31.

156.   Veldman, J.B.P., Mulder, L.J.M., Mulder, G., and VanDerHeide, D. "Attention, Effort, and Sinus Arrhythmia: How Far Are We?" *Psychophysiology of Cardiovascular Control*, eds. J.F. Orlebeke, G. Mulder, and L.J. VanDoornen. New York, NY: Plenum Press, 1985, pp. 407-424.

157.   Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., and Alkon, D.L. "Acceleration the Convergence of the Backpropagation Method," *Biological Cybernetics*, Vol 59, 1988, pp. 257-263.

158.   Waibel, A., Hanazawa, T., Hinton, G., Shikano, and Lang, K.J. "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol 37, No 3, March 1989, pp. 328-339.

159.   Walter, W.G., Cooper, R., Aldridge, V.J., McCallum, W.C., and Winter, A.L. "Contingent Negative Variation: An Electrical Sign of Sensory Motor Association and Expectancy in the Human Brain," *Nature*, Vol 203, 1964, pp. 380-384.

160.   Weiss, S.M. and C.A. Kulikowski. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA: Morgan Kaufman, 1991.

161.   Werbos, P.J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. dissertation, Harvard University, Cambridge, Mass., 1974.

162.   White, H. *Artificial Neural Networks Approximation and Learning Theory.* Cambridge, MA: Blackwell Publishers, 1993.

163.   White, H. "Some Asymptotic Results for Learning in Single Hidden-Layer Feedforward Network Models," *Journal of the American Statistical Association*, Vol 84, No 408, Dec 1989, pp. 1003-1013.

164.   Wientjes, C.J.E. "Respiration in Psychophysiology: Methods and Applications," *Biological Psychology*, Vol 34, 1992, pp. 179-204.

165.   Wientjes, C.J.E. *Respiration and Stress*, Ph.D. Dissertation, University of Tilburg, Tilburg, Netherlands, 1993.

166.   Williams, R.J. and D. Zipser. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, Vol 1, 1989, pp. 270-280.

167.   Williams, R.J. and D. Zipser. "Experimental Analysis of the Real-Time Recurrent Learning Algorithm," *Connection Science*, Vol 1, No 1, 1989, pp. 87-111.

168.   Williges, R.C. and W.W. Wierwille. "Behavioral Measures of Aircrew Mental Workload," *Human Factors*, Vol 21, 1979, pp. 549-574.

169.   Wilson, G.F. "Applied Use of Cardiac and Respiration Measures: Practical Considerations and Precautions," *Biological Psychology*, Vol 34, 1992, pp. 163-178.

170.   Wilson, G.F. "Air-to-Ground Training Missions. A Psychophysiological Workload Analysis," *Ergonomics*, Vol 36, No 9, 1993, pp. 1071-1087.

171.   Wilson, G.F. Preface to *Biological Psychology*, Vol 40, 1995, pp. vii-viii.

172.   Wilson, G.F. "Workload Assessment Monitor," *Proceedings of the Human Factors Society*, 1994, pp. 944.

173.   Wilson, G.F. and F.T. Eggemeier. "Psychophysiological Assessment of Workload in Multi-Task Environment," *Multiple-Task Performance*, Damos, D.L. (ed.). London, UK: Taylor & Francis, 1991, pp. 229-260.

174.   Wilson, G.F. and F. Fisher. "Cognitive Task Classification Based Upon Topographic EEG Data," *Biological Psychology*, Vol 40, 1995, pp. 239-250.

175.   Wilson, G.F., Fullenkamp, P., and Davis, I. "Evoked Potential, Cardiac, Blink, and Respiration Measures of Pilot Workload in Air-to-Ground Missions," *Aviation, Space, and Environmental Medicine,* Feb 1994, pp. 100-105.

176. Wilson, G.F., and O'Donnell, R.D. "Steady-State Evoked Responses: Correlations with Human Cognition," *Psychophysiology*, Vol 23, 1986, pp. 57-61.

177. Wilson, G.F., Swain, C.R., and Brookings, J.B. *Workload Related Changes in Eye, Cardiac, Respiratory and Brain Activity During Simulated Air Traffic Control*, AL/CF-TR-1995-0156, 1995.

Vita

Captain Kelly A. Greene was born Kelly A. Kratochvil on 2 Oct 1968 in Oakland, California. In 1986, Kelly graduated from Glen A. Wilson High School in Hacienda Heights, California, and entered the United States Air Force Academy. In 1990, she graduated from the United States Air Force Academy with a B.S. in Operations Research and was commissioned as a Second Lieutenant in the United States Air Force. After graduating from the Academy, Kelly married Christopher D. Greene of Indio, California. Shortly thereafter, she entered graduate school at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. In 1992, Kelly graduated from the Air Force Institute of Technology with a M.S. in Operations Research and then reported for her first duty assignment as a Measurement and Signature Intelligence (MASINT) Project Engineer at the Foreign Aerospace Science and Technology Center, Wright-Patterson Air Force Base, Ohio. In 1993, she reported for duty as a Flight Test Analyst at the 18th Flight Test Squadron, Hurlburt Field, Florida. Kelly returned to the Air Force Institute of Technology in 1995 to pursue a Ph.D. in Operations Research. After graduation from the Air Force Institute of Technology, Kelly will begin a year of Flight Test Engineer training at the United States Air Force Test Pilot School, Edwards Air Force Base, California. Kelly's hobbies include flying, SCUBA diving, and snow skiing.

Permanent address: c/o H. Edward Shaffer
                     18132 Hearth Drive
                     Fountain Valley CA 92708

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1998	3. REPORT TYPE AND DATES COVERED Ph.D. Dissertation

4. TITLE AND SUBTITLE
FEATURE SALIENCY IN ARTIFICIAL NEURAL NETWORKS WITH APPLICATION TO MODELING WORKLOAD

5. FUNDING NUMBERS

6. AUTHOR(S)
Kelly A. Greene, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology (AFIT)
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DS/ENS/98-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. John F. Tangney	Dr. Glenn F. Wilson
AFOSR/NL	AFRL/HECP
110 Duncan Ave, Suite B115	2255 H Street
Bolling AFB DC 20322-0001	Wright-Patterson AFB OH 45433-7022
(202) 767-8075 (DSN 297)	(937) 255-8748 (DSN 785)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Advisor was Dr. Kenneth W. Bauer, Jr., of AFIT/ENS (937) 255-6565 x4328 (DSN 785)
kbauer@afit.a.fmil

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This dissertation research extends the current knowledge of feature saliency in artificial neural networks (ANN). Feature saliency measures allow for the user to rank order the features based upon the saliency, or relative importance, of the features. Selecting a parsimonious set of salient input features is crucial to the success of any ANN model. In this research, several methodologies were developed using the *Signal-to-Noise Ratio (SNR) Feature Screening Method* and its associated *SNR Saliency Measure* for selecting a parsimonious set of salient features to classify pilot workload in addition to air traffic controller workload. Candidate features were derived from electroencephalography (EEG), electrocardiography (EKG), electro-oculography (EOG), and respiratory gauges. In addition, a new saliency measure was developed that can account for time in Elman Recurrent Neural Networks (RNN). This *Partial Derivative Based Spatial-Temporal Saliency Measure* is used via a *Spatial-Temporal Feature Screening Method* for selecting a parsimonious set of salient features in both time and space. Finally, a technique for investigating the memory capacity of an Elman RNN was developed.

14. SUBJECT TERMS
Feature saliency, Feature selection, Neural networks, Spatial-temporal, Workload

15. NUMBER OF PAGES
296

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94